SECOND EDITION

FUNDAMENTALS OF SUPPLY CHAIN THEORY

Lawrence V. Snyder | Zuo-Jun Max Shen

with website

WILEY

Fundamentals of Supply Chain Theory

Fundamentals of Supply Chain Theory

Lawrence V. Snyder
Lehigh University

Zuo-Jun Max Shen
University of California, Berkeley

Second Edition

WILEY

This edition first published 2019

Edition History
John Wiley & Sons, Inc. (1e, 2011)

Registered Office
John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, USA

Editorial Office
111 River Street, Hoboken, NJ 07030, USA

For details of our global editorial offices, customer services, and more information about Wiley products visit us at www.wiley.com.

Wiley also publishes its books in a variety of electronic formats and by print-on-demand. Some content that appears in standard print versions of this book may not be available in other formats.

Library of Congress Cataloging-in-Publication Data

Names: Snyder, Lawrence V., 1975- author. | Shen, Zuo-Jun Max, 1970- author.
Title: Fundamentals of supply chain theory / Lawrence V. Snyder, Lehigh University, Zuo-Jun Max Shen, University of California, Berkeley.
Description: Second edition. | Hoboken, New Jersey : John Wiley & Sons, Inc., [2019] | Includes bibliographical references and index. |
Identifiers: LCCN 2019015579 (print) | LCCN 2019018047 (ebook) | ISBN 9781119024866 (Adobe PDF) | ISBN 9781119024972 (ePub) | ISBN 9781119024842 (hardcover)
Subjects: LCSH: Business logistics.
Classification: LCC HD38.5 (ebook) | LCC HD38.5 .S6256 2020 (print) | DDC 658.701–dc23
LC record available at https://lccn.loc.gov/2019015579

Cover Design: Wiley
Cover Image: © RomoloTavani/Getty Images

Set in 10/12pt NimbusRom by Lawrence V. Snyder and Zuo-Jun Max Shen

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

To Suzanne, Coralie, and Matilda
–L.V.S.

To Irene, Michelle, and Jeffrey
–Z.-J.M.S.

CONTENTS IN BRIEF

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

PREFACE

Goals of This Book

In the past few decades, the study of supply chain management has evolved into a cohesive body of knowledge—not merely a haphazard collection of models, algorithms, and theorems, but a rich theory whose components intersect and inform each other. We wrote this book to help codify the foundations of this emerging supply chain theory and to demonstrate how recent developments build upon the classical models. Our focus is primarily on the seminal models and algorithms of supply chain theory—the building blocks that underlie much of the supply chain literature. We believe that an understanding of these models provides researchers with a sort of guidebook to the literature, as well as a toolbox to draw from when developing new models. We also discuss some more recent models that demonstrate how the classical models can be extended and applied in richer settings. These models provide graduate students and other new researchers in the field with some examples of the trajectory of research on supply chain theory—how the building blocks can be assembled to create something more complex, interesting, or useful.

Studying supply chain theory as a whole allows us the luxury of gaining some perspective on the field, a perspective that is not always evident when we immerse ourselves deeply in the literature on a particular topic. To that end, wherever possible, we have attempted to highlight the connections among supply chain models—for example, the conceptual similarities among different supply chain pooling models, the ways that inventory and location models can be combined, or the ways that inventory theory interacts with game theory to produce supply chain coordination models.

Who Should Read This Book

This book was written for anyone who is interested in quantitative approaches for studying supply chains. This includes people from a wide range of disciplines, such as industrial engineering/operations research, mathematics, management, economics, computer science, and finance. This also includes students (primarily graduate students), faculty, researchers, and practitioners of supply chain theory. And it includes scholars who are new to supply chain theory and want a gentle but rigorous introduction to it, or scholars who are well versed in the field and want a refresher or a reference for the seminal models. Finally, since you are holding this book, it most likely includes you.

One of the hallmarks—and, in our opinion, the great pleasures—of supply chain theory is that it makes use of a wide variety of the tools of operations research, mathematics, and computer science. In this book, you will find mathematical programming models (linear, integer, nonlinear, conic, stochastic, robust), duality theory, optimization techniques (Lagrangian relaxation, column generation, dynamic programming, line search, plus optimization by calculus and finite differences), heuristics and approximations, probability, stochastic processes, game theory, combinatorics, simulation, and complexity theory.

To make use of this book, you need not be an expert in all of these. (We are not.) We assume that you are familiar with basic optimization theory—that you know how to formulate a linear program and its dual, that you know how branch-and-bound works, and that you can perform a simple line search method such as bisection search. We also assume that you understand probability distributions and know how to compute expectations of random variables and functions thereof. We assume that your calculus is in good working order, that you can compute derivatives and integrals, including ones that involve multiple variables or other derivatives or integrals. We assume you have met Markov chains before, but we don't require you to remember much about them. For just about everything else, we will start from the ground up and tell you (or remind you of) what you need to know in order to understand the topic at hand. For some topics, you will find a useful reference in Appendix C, which lists formulas for calculating expectations, loss functions, geometric series, and some tricky derivatives and integrals. Because Lagrangian relaxation and column generation play a role in several chapters of this book, we have included a brief primer on those topics in Appendix D.

Probably the single most important prerequisite for this book is a high level of general mathematical maturity. We discuss a lot of mathematical proofs, and ask you to write your own in the homework problems. If you do not have much experience in this area, you may find the proofs to be the most challenging aspect of this book. To help you out, we have included in Appendix B a short guide to proof-writing. We hope this appendix will familiarize you with some of the basic principles of proof-writing, as well as some of the finer points of proof style and syntax. But, proof-writing is perhaps more art than science, and the appendix will only get you so far. You will learn to be a good proof-writer mainly by practicing the craft.

Organization of This Book

Our intention in writing this book was to cover a broad range of topics in supply chain theory, even if that meant that we could not cover some topics as deeply as we might have liked. Most of the material in this book is derived from earlier papers, and of course we have cited those papers carefully so that readers can delve deeper into any topics they wish. We have also cited important related references, and review articles where possible, so that readers can find more information about topics that interest them.

Most of this book (Chapters 2–12) deals with *centralized* supply chain models, in which all of the decision variables are under the control of a single decision-maker. Most classical supply chain models, such as those for optimizing inventories and facility locations, are centralized models. In contrast, the *decentralized* models of Chapters 13–15 involve multiple parties with independent, conflicting objectives and the autonomy to choose their decision variables to optimize those objectives. The bullwhip effect (Chapter 13) is an example of a result of this decentralization, while the models of Chapters 14 and 15 discuss strategies for mitigating the negative financial effects of decentralization.

This chapters of this book are as follows:

- Chapter 1 ("Introduction") gives an overview of supply chain management and defines terms that we will use throughout the book.

- Chapter 2 ("Forecasting and Demand Modeling") discusses classical and machine-learning–based forecasting methods, as well as three approaches—the Bass diffusion model, leading indicators, and choice models—that have been used more recently to predict demand. We refer to these latter approaches as "demand modeling" to differentiate them from classical forecasting techniques and to emphasize the fact that they aim to provide a model of the demand itself and not merely of its statistical properties.

- We discuss classical single-location inventory models in Chapters 3 ("Deterministic Inventory Models"), 4 ("Stochastic Inventory Models: Periodic Review"), and 5 ("Stochastic Inventory Models: Continuous Review"). For most of these models, we discuss how to formulate the objective function as well as how to optimize it—exactly or heuristically, in closed form or using algorithms—by our choice of inventory parameters. We also explore the theoretical properties of some of these models, including the optimality of inventory policies and the worst-case performance of heuristics.

- In Chapter 6 ("Multiechelon Inventory Models"), we discuss multiechelon inventory models, including both stochastic-service models (including the Clark–Scarf model for serial systems and the Shang and Song approximation) and guaranteed-service models (also known as strategic safety stock placement problems).

- Chapter 7 ("Pooling and Flexibility") discusses risk pooling, as well as other techniques, such as postponement, transshipments, and process flexibility, that can provide similar pooling benefits.

- In Chapter 8 ("Facility Location Models"), we turn our attention to facility location models. We present the classical uncapacitated fixed-charge location problem (UFLP) in some detail, including its formulation as an integer programming problem and its solution by Lagrangian relaxation. We then discuss other classical location models such as the p-median problem and covering models, as well as stochastic versions of the UFLP. Finally, we cover network design problems, including both problems in which we make yes/no decisions on the nodes and those in which we do the same for the arcs.

- In Chapter 9 ("Supply Uncertainty"), we consider randomness in the availability or quantity of supply and develop models for coping with this uncertainty in inventory and facility location models.
- Chapter 10 ("The Traveling Salesman Problem") discusses perhaps the most famous supply chain problem, the traveling salesman problem (TSP). We discuss both exact and heuristic solution methods for the TSP, as well as theoretical properties of the model and the algorithms. We conclude with a digression on TSP "world records."
- In Chapter 11 ("The Vehicle Routing Problem"), we extend the TSP to consider the more practical problem of routing multiple vehicles simultaneously to deliver to many customers, a problem known as the vehicle routing problem (VRP). We present algorithms, focusing mainly on heuristics for this very difficult computational problem. We discuss theoretical properties of the problem, as well as some of the many extensions that have been proposed to add more practical features to the classical model.
- Chapter 12 ("Integrated Supply Chain Models") discusses models that combine multiple types of models discussed earlier in the book. In particular, we include location–inventory, location–routing, and inventory–routing models.
- In Chapter 13 ("The Bullwhip Effect"), we discuss a phenomenon of demand variability amplification known as the bullwhip effect. The bullwhip effect can occur because of irrational or suboptimal behavior on the part of supply chain managers, but it can also occur as the result of rational, optimizing behavior. We cover mathematical models for proving that the bullwhip effect occurs as a result of the latter type.
- When supply chain partners each optimize their own objective functions, they typically arrive at solutions that are suboptimal from the point of view of the total supply chain. In Chapter 14 ("Supply Chain Contracts"), we discuss contracts that achieve coordination within a supply chain made up of individual players with differing objectives.
- Chapter 15 ("Auctions") introduces mathematical models for auctions, which are frequently used to set prices within supply chains. Auctions can be thought of as another way to mitigate the effects of decentralized decision-making and to bring supply chains into closer coordination.
- Chapter 16 ("Applications of Supply Chain Theory") explores three non-supply-chain fields in which supply chain theory has been widely applied: electricity systems, health care, and public sector operations. In each of these topics, we cover a few (typically more recent) models that directly apply the tools you will have learned earlier in the book. Our aim is to demonstrate the application of supply chain theory, now that you have mastered its methodologies.
- The book concludes with four appendices. Appendix A contains homework problems whose solutions use material from multiple chapters. Appendix B provides a short

primer on how to write mathematical proofs. Appendix C lists helpful formulas that are used throughout the book. Appendix D gives a brief overview of Lagrangian relaxation and column generation.

The material in this book can accommodate a good deal of reordering and omission by the instructor. The only real exception is the inventory-theoretic material (Chapters 3–6), which is at the core of much of the subsequent material in the book and therefore should be covered early on. However, not all of the material in the inventory chapters is used elsewhere, and much of it can be skipped if desired. A bare-bones treatment of the essential inventory topics would include Section 3.2 on the EOQ model, Section 4.3.2 on the newsvendor problem, and Section 5.1 on (r, Q) policies—and even this material could be omitted for students who are already familiar with it. In addition, the material in Section 9.6 and Chapter 12 relies on the facility location chapter (Chapter 8), primarily Section 8.2. And of course, Chapter 16, on applications of supply chain theory, draws on material from throughout the book. Other than these, there are no precedence constraints regarding the sequence of material covered, and the instructor is free to rearrange the topics according to his or her preferences, interests, and expertise, as well as those of the students.

The final section of each chapter (except Chapter 1) contains a case study, a new feature in the second edition. The case studies are drawn from the journal *Interfaces* (now called the *INFORMS Journal on Applied Analytics*). Each case study illustrates an application by a real company or organization of the ideas discussed in the chapter. The case studies show the reader how supply chain theory can be applied, sometimes as-is and sometimes with substantial modifications, to solve real-world problems with significant impact.

We have adapted the original notation for the models discussed in the case studies, in order to be consistent with the rest of the book. In some cases we have also simplified or made other minor modifications to the models, while striving to maintain the main ideas of the original models. Each case study gives some basic facts about the company involved—for example, its ranking within its industry. We have attempted to update these facts where possible, but in general the reader should assume the facts were correct at the time that the original *Interfaces* article was published, if not still true today, even if we use the present tense in stating them.

Each of the chapters (again, except Chapter 1) is followed by a set of homework problems, and Appendix A presents problems that use material from multiple chapters. The problems challenge readers to understand, interpret, and extend the models and algorithms discussed in the text. Some of them involve simply applying the models and algorithms presented in the book as-is. Most of them, however, ask the reader to prove theorems, develop models, or somehow explore the material more deeply than it is covered in the chapters. Some of the problems require data sets that are too large to include in the text itself. These data sets are posted on the web site for this book. Where relevant, citations to the original sources for homework problems are given in the solutions, rather than in the problems themselves.

The book's web site also contains a list of errata. If you find errors not contained on this list, please e-mail the authors, whose contact information can also be found on the site.

New in the Second Edition

The second edition of *Fundamentals of Supply Chain Theory* is nearly twice as long as the first. The book has been revised from beginning to end. We have added three entirely new chapters, on the TSP, the VRP, and applications of supply chain theory. The inventory chapters have been reorganized and significantly expanded, as has the facility location chapter. We have rearranged the material on risk pooling and supply uncertainty into (we feel) more logical groupings. Other new topics include machine learning models for forecasting (Section 2.4), a multisupplier inventory model with supply uncertainty (Section 9.4), a conic optimization approach for the LMRP (Section 12.2.8), location–routing and inventory–routing models (Sections 12.3 and 12.4), a game-theoretic analysis of the VCG auction (Section 15.4.3), and a primer on column generation (Section D.2).

The end-of-chapter case studies are a new feature for the second edition. We have added nearly 200 new homework problems and over 60 new worked examples. We redesigned all of the figures for improved clarity and have added 140 new ones. The algorithm pseudocode has been updated to a more modern format, and the index is now more comprehensive.

Resources for Instructors

We have developed the following resources to assist instructors:

- An instructor's manual containing full solutions to the homework problems
- PowerPoint slides for in-class presentation of the book material
- In-depth MATLAB coding assignments so that students can implement the models and algorithms discussed in the book

These resources are available to verified instructors via links on the book's web site.

Acknowledgments

We owe a debt of gratitude to many people for many reasons. First, we wish to thank Mindy Okura-Marszycki, Susanne Steitz-Filler, Vini Premkumar, Kathleen Pagliaro, Nithya Sechin, Melissa Yanuzzi, Jackie Palmieri, and the rest of the editorial team at Wiley, for championing the book and bringing it to fruition.

We would like to thank our professors at Northwestern University who taught us while we were graduate students there. Special thanks go to Mark Daskin and David Simchi-Levi, who served as our advisors and mentors. The models in Sections 9.6 and 12.2, in particular, are the results of our collaborations with Mark. Both Mark and David are outstanding researchers, excellent teachers, and generous, supportive advisors—not to mention accomplished textbook authors—and we would not be the professors we are without them.

We thank our colleagues at Lehigh and Berkeley, as well as our current and former Ph.D. students, especially Zümbül Atan, Gang Chen, Leon Chu, Tingting Cui, Tianhu Deng, Lin He, Çağrı Latifoğlu, Shan Li, Yinan Liu, Ho-Yin Mak, Mohsen Moarefdoost, Lian Qi, Ying Rong, Amanda Schmitt, Ye Xu, and Lezhou Zhan, for contributing to this book through their research collaborations (some of which are reflected in the material in this book) and the many productive discussions we have had with them about research and teaching.

This book emerged from lecture notes we developed for our graduate-level supply chain courses at Lehigh and Berkeley. Many students suffered through the early versions of these notes. Their questions, suggestions, and confused faces helped us find and correct errors and improve the exposition throughout the book. Tingting Cui, Tao Li, Ho-Yin Mak, Scott DeNegre, Kewen Liang, Gokhan Metan, Cory Minglegreen, Jack Oh, Jim Ostrowski, Ye Xu, Mertcan Yetkin, and Hua Zhong, among others, made insightful comments that resulted in a better explanation, an interesting new homework problem, or an elegant solution to a problem.

Tolga Seyhan provided invaluable assistance with the preparation of the first edition of this book, crafting many of the figures, writing the index, assembling references, and lending us his impeccable attention to detail. We also thank Pete Ferrari for helping us build BibTeX databases, Amy Hendrickson of TeXnology, Inc. for sharing her expertise in all things LaTeX, and Andrew Ross for his patient answers to questions about stochastic processes. Karen Smilowitz contributed suggestions, feedback, and the occasional homework problem.

Reza Nazari and Afshin Oroojlooy contributed greatly to the second edition of the book through their careful reading of the manuscript, catching many mistakes and offering valuable feedback. Wancheng Feng, Sheng Liu, and Yuli Zhang also offered great help with the preparation of the second edition. Bill Cook of the University of Waterloo generously provided us with TSP data sets and insights. And many people wrote to us to point out errors in the first edition; we thank especially Gil Souza, as well as Onur Babat, Mark Bai, Ory Ball, Ali Diabat, Bisheng Du, Mohammad Ghuloum, Jian Luo, Josh Margolis, Kwami Senam Sedzro, Jieli Tian, Siyu Yang, Zhu Yang, Rui Yu, Dell Zhang, and others too numerous to mention.

Finally, and most importantly, we thank our families—Suzanne, Irene, Matilda, Michelle, Coralie, and Jeffrey—and our extended families for their support, love, encouragement, and guidance as we wrote, and revised, this book.

About the Companion Website

This book is accompanied by a companion website:

www.wiley.com/go/Snyder/SupplyChainTheory

The companion website consists of a student website and an instructor website, and contains:

- Student Website
  - Data Sets
  - Errata
- Instructor Website
  - Instructor's Manual
  - PowerPoint Slides
  - MATLAB Coding Assignments

CHAPTER 1

INTRODUCTION

1.1 THE EVOLUTION OF SUPPLY CHAIN THEORY

The field of supply chain management arose from managers' recognition that buying, selling, manufacturing, assembling, warehousing, transporting, and delivering goods—that is, the activities of a supply chain—are expensive endeavors, and that careful attention to how these activities are carried out may reduce their cost. Supply chains used to be viewed, at least by some managers, as "necessary evils." As a result, the mindset for supply chain managers revolved around reducing costs, by reducing inventory levels, taking advantage of economies of scale in shipping, optimizing network designs, reducing volatility in demands, and so on. By and large, these improvements were invisible to companies' customers, provided that they did not result in longer lead times, more frequent stockouts, or other degraded service.

By the end of the last century, however, the purpose of the supply chain had begun to change as some firms discovered that supply chains could be a source of competitive advantage, rather than simply a cost driver. For example, Dell demonstrated that, through excellent supply chain management, it could deliver computers—fully customized to the buyer's specifications—just a few days after they were ordered. In doing so, it shattered the existing paradigm for computer purchases, in which consumers could choose from only a limited number of preconfigured options. Similarly, Walmart showed that, by operating an extremely high-volume supply chain, it could land products on shelves for less money per item. As a result, Walmart offered its customers a high level of product availability and

Fundamentals of Supply Chain Theory, Second Edition. Lawrence V. Snyder and Zuo-Jun Max Shen.

Companion website: www.wiley.com/go/Snyder/SupplyChainTheory

low prices, and this combination ushered the company to its place as the world's largest retailer. Amazon built a supply chain that is not only quick and reliable, but also feature-rich, offering users varied shipping options, convenient tracking tools, and flexible return policies. This expansive supply chain has allowed Amazon to overcome consumers' desire for instant gratification and their preference for seeing and touching products before they buy them.

Just as the *practice* of supply chain management has come into its own, so, we would argue, has the *study* of supply chain management. In the past several decades, a huge number of papers have been published that introduce mathematical models for evaluating, analyzing, and optimizing supply chains. Supply chain management has become one of the most popular applications of operations research (OR), and one of its greatest success stories. But recently, the mathematical study of supply chains has begun to be viewed not simply as an application area for OR tools, but rather as a methodological area, capable of standing on its own two feet, with its own tools and theory. These tools are now themselves starting to be applied, not just to supply chains, but to health care, energy, humanitarian relief, the service sector, and other industries. This emerging supply chain theory is the subject of this book.

Although the models and algorithms in this book are most commonly applied to traditional, private-sector supply chains, many can be applied to new kinds of supply chains, and even to areas we might not think of as supply chains. Understanding the building blocks of traditional supply chains will prepare you to understand more recent applications of supply chain theory. The final chapter of this book is devoted to exploring how the tools of supply chain theory are used in a few of these application areas—electricity systems, health care, and public sector operations.

1.2 DEFINITIONS AND SCOPE

The term *supply chain management* is difficult to define, and its definition has changed over time as the purposes and components of supply chains have evolved. Perhaps the most authoritative definition comes from the Council of Supply Chain Management Professionals (CSCMP), who define supply chain management as follows:

> Supply chain management encompasses the planning and management of all activities involved in sourcing and procurement, conversion, and all logistics management activities. Importantly, it also includes coordination and collaboration with channel partners, which can be suppliers, intermediaries, third party service providers, and customers. In essence, supply chain management integrates supply and demand management within and across companies. (Council of Supply Chain Management Professionals 2018b)

In the interest of keeping things a little simpler, we offer the following definitions:

> A *supply chain* consists of the activities and infrastructure whose purpose is to move products from where they are produced to where they are consumed. *Supply chain management* is the set of practices required to perform the functions of a supply chain and to make them more efficient, less costly, and more profitable.

Supply chain management costs firms nearly US$1.5 trillion per year in the United States alone, representing nearly 8% of gross domestic product (GDP) (Council of Supply Chain Management Professionals 2018a). These practices include a huge range of tasks, such as forecasting, production planning, inventory management, warehouse location, supplier selection, procurement, and shipping. Mathematical models have been developed to analyze and optimize each of these practices, and these models are the primary focus of this book.

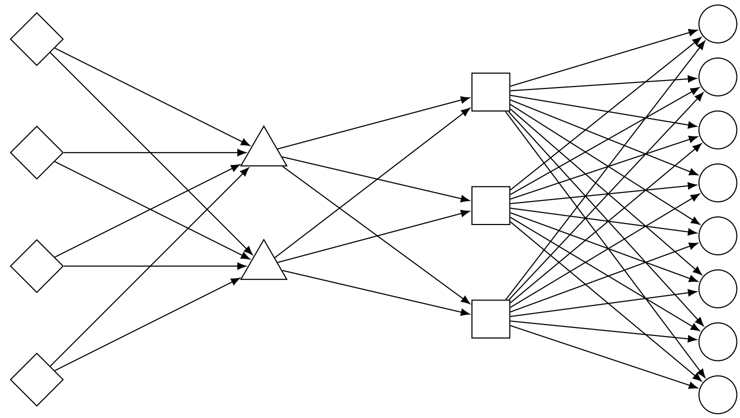


Figure 1.1 Schematic diagram of supply chain network.

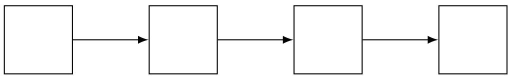


Figure 1.2 Supply "chain."

The terms "logistics" and "logistics management" are closely related to "supply chain management," and it can be difficult to draw a clear distinction. Some companies use "logistics" to refer to the physical movement of goods; "supply chain management" includes logistics, as well as nonmovement activities such as inventory management and procurement. For other companies, "logistics" means functions carried out by the company itself, while "supply chain management" includes activities it conducts with partners, suppliers, and customers. Often, though, the two terms are used more or less interchangeably.

Supply chains are often represented graphically as a schematic *network* that illustrates the relationships between its elements. (See Figure 1.1.) Each vertical "level" of the supply chain (suppliers, plants, etc.) is called an *echelon*. A location in the network is referred to as a *stage* or *node*. The links between stages represent some type of flow—typically, the flow of goods, but sometimes the flow of information or money. The portion of the supply chain from which products originate (the left-hand portion in Figure 1.1) is referred to as *upstream*, while the demand end is referred to as *downstream*.

Actually, the phrase "supply chain" is a bit of a misnomer, since "chain" implies a linear system similar to the one pictured in Figure 1.2. In this system, sometimes referred to as a *serial system*, each echelon has only a single stage. But today's supply chains more closely resemble the complex network in Figure 1.1; each echelon may have dozens, hundreds, or even thousands of nodes. (Nevertheless, we will often study serial systems of the type pictured in Figure 1.2. Even more frequently, we will study single-stage systems.)

The models we study generally try to find the least-cost or greatest-profit solution that satisfies some constraints. For example, a firm might want to choose warehouse locations to minimize transportation costs, subject to the constraint that every customer must be served. Or it might want to decide how much inventory should be stored at a given warehouse in order to minimize the cost of holding inventory, subject to a “service level” constraint that requires a certain percentage of customer orders to be satisfied on time. Or it might want to design a contract with its supplier to maximize its own profit, or that of the supply chain as a whole.

The ideal supply chain management model would *globally* optimize every aspect of the supply chain, but such a model is impossible both because of the difficulties in modeling some aspects of the supply chain mathematically and because the resulting model would be too large and complex to solve. Instead, supply chain models typically focus on *local* optimization of one element of the supply chain, or on the integration of two or more aspects of the supply chain, generally in less detail.

1.3 LEVELS OF DECISION-MAKING IN SUPPLY CHAIN MANAGEMENT

It is convenient to think about three levels of supply chain management decisions: strategic, tactical, and operational.

- *Strategic* aspects of the supply chain involve decisions that take effect over a long time horizon, typically years or decades. These aspects have a major impact on all functions of the firm. Examples include locations and sizes of warehouses, locations and capabilities of factories, and contracts with suppliers.

- *Tactical* aspects of the supply chain involve decisions over a moderate time horizon like months. Tactical decisions can be changed periodically but generally with some difficulty. Examples include assignments of customers to warehouses and inventory replenishment policies at warehouses.

- *Operational* aspects of the supply chain occur over short planning horizons such as days or weeks, during which policies must be executed but cannot be changed. Examples include filling customer orders and routing of delivery vehicles.

The models in this book are concerned with all three levels of decisions. For example, the facility location models of Chapters 8 and 12 are strategic, the inventory models of Chapters 3–6 are tactical, and the routing models of Chapters 10 and 11 are operational.

CHAPTER 2

FORECASTING AND DEMAND MODELING

2.1 INTRODUCTION

Demand forecasting is one of the most fundamental tasks that a business must perform. It can be a significant source of competitive advantage by improving customer service levels and by reducing costs related to supply–demand mismatches. In contrast, biased or otherwise inaccurate forecasting results in inferior decisions and thus undermines business performance.

For example, the toy retailer Toys "R" Us made a huge mistake in demand forecasting for the 2015 Christmas season. For several days, the actual number of online orders was more than twice the company's forecasts, and the company's distribution centers were overwhelmed. As a result, the company was forced to throttle demand by terminating some online sales, resulting in lower demand and lower revenue (Ziobro 2016).

The goal of the forecasting models discussed in this chapter is to estimate the quantity of a product or service that consumers will purchase. Most classical forecasting techniques involve time-series methods that require substantial historical data. Some of these methods are designed for demands that are stable over time. Others can handle demands that exhibit trends or seasonality, but even these require the trends to be stable and predictable. However, products today have shorter and shorter life cycles, in part driven by rapid technology upgrades for high-tech products. As a result, firms have much less historical data available to use for forecasting, and any trends that may be evident in historical data may be unreliable for predicting the future.

Fundamentals of Supply Chain Theory, Second Edition. Lawrence V. Snyder and Zuo-Jun Max Shen.

Companion website: www.wiley.com/go/Snyder/SupplyChainTheory

In this chapter, we first discuss some classical methods for *forecasting* demand, in Sections 2.2 and 2.3. Next, in Section 2.4, we discuss more recent approaches to forecasting demand using machine learning when we have large quantities of historical data available. In Sections 2.5–2.8, we discuss several methods that can be used to predict demands for new products or products that do not have much historical data. To distinguish these methods from classical time-series–based methods, we call them *demand modeling* techniques.

The methods that we discuss in this chapter are quantitative. They all involve mathematical models with parameters that must be calibrated. In contrast, some popular methods for forecasting demand with little or no historical data, such as the *Delphi method*, rely on experts' qualitative assessments or questionnaires to develop forecasts.

Demand processes may exhibit various forms of nonstationarity over time. These include the following:

- *Trends*: Demand consistently increases or decreases over time.
- *Seasonality*: Demand shows peaks and valleys at consistent intervals.
- *Product life cycles*: Demand goes through phases of rapid growth, maturity, and decline.

Moreover, demands exhibit *random error*—variations that cannot be explained or predicted—and this randomness is typically superimposed on any underlying nonstationarity.

2.2 CLASSICAL DEMAND FORECASTING METHODS

Classical forecasting methods use prior demand history to generate a forecast. Some of the methods, such as moving average and (single) exponential smoothing, assume that past patterns of demand will continue into the future, that is, no trend is present. As a result, these techniques are best used for mature products with a large amount of historical data. On the other hand, regression analysis and double and triple exponential smoothing can account for a trend or other pattern in the data. We discuss each of these methods next.

In each of the models that follow, we use $D_1, D_2, \ldots, D_t, \ldots$ to represent the historical demand data, i.e., the realized demands in periods 1, 2, ..., t, ... We also use y_t to denote the forecast of period t's demand that is made in period $t-1$.

2.2.1 Moving Average

The *moving average* method calculates the average amount of demand over a given interval of time and uses this average to predict the future demand. As a result, moving average forecasts work best for demand that has no trend or seasonality. Such demand processes can be modeled as follows:

$$D_t = I + \epsilon_t, \tag{2.1}$$

where I is the mean or "base" demand and ϵ_t is a random error term.

A moving average forecast of order N uses the N most recent observed demands. The forecast for the demand in period t is simply given by

$$y_t = \frac{1}{N} \sum_{i=t-N}^{t-1} D_i. \tag{2.2}$$

Table 2.1 Monthly historical demand of books and CDs for Examples 2.1–2.5.

	Demand (Thousands)		
Month	*An Inventory Story*	*The TSP Mystery*	CDs
1	10.61	12.61	10.21
2	12.01	16.01	23.01
3	9.77	15.77	10.97
4	10.19	18.19	14.59
5	9.44	19.44	29.44
6	11.40	23.40	16.80
7	9.66	23.66	18.86
8	9.90	25.90	38.90
9	9.01	27.01	18.61
10	10.20	30.20	24.20
11	10.90	32.90	48.90
12	8.98	32.98	22.78

That is, the forecast is simply the arithmetic mean of the previous N observations. This is known as a *simple moving average forecast of order* N.

A generalization of the simple moving average forecast is the *weighted moving average*, which allows each period to carry a different weight. For instance, if more recent demand is deemed more relevant, then the forecaster can assign larger weights to recent demands than to older ones. If w_i is the weight placed on the demand in period i, then the weighted moving average forecast is given by

$$y_t = \frac{\sum_{i=t-N}^{t-1} w_i D_i}{\sum_{i=t-N}^{t-1} w_i}. \tag{2.3}$$

Typically, the weights decrease by 1 in each period: $w_{t-1} = N$, $w_{t-2} = N - 1$, ..., $w_{t-N} = 1$.

☐ **EXAMPLE 2.1**

A book store has historical demand data for the book *An Inventory Story* for the past 12 months, as shown in Table 2.1. From a quick look, it is clear that the demand is relatively stable, fluctuating around the value 10, which makes it suitable for the moving average method. Suppose the book store manager wants to predict the demand of this book for the next month. Using an order of $N = 5$, the forecast is given by

$$y_{13} = \frac{D_8 + D_9 + D_{10} + D_{11} + D_{12}}{5} = 9.80.$$

☐

2.2.2 Exponential Smoothing

Exponential smoothing is a technique that uses a weighted average of all past data as the basis for the forecast. It gives more weight to recent information and smaller weight to observations in the past. Single exponential smoothing assumes that the demand process is stationary. Double exponential smoothing assumes that there is a trend, while triple exponential smoothing accounts for both trends and seasonality. These methods all require user-specified parameters that determine the relative weights placed on recent and older observations when predicting the demand, trend, and seasonality. These three weights are called, respectively, the *smoothing factor*, the *trend factor*, and the *seasonality factor*. We discuss each of these three methods next.

2.2.2.1 Single Exponential Smoothing Define $0 < \alpha \leq 1$ as the smoothing constant. Then, we can express the current forecast as the weighted average of the previous forecast and most recently observed demand value:

$$y_t = \alpha D_{t-1} + (1 - \alpha) y_{t-1}. \tag{2.4}$$

Note that α is the weight placed on the demand observation and $1 - \alpha$ is the weight placed on the last forecast. Typically, we place more weight on the previous forecast, so α is closer to 0 than to 1.

Since each forecast depends on the previous forecast, we need a way to get the process started. One simple way to do this is to set $y_1 = D_1$. Note that this method requires one historical demand observation D_1; the first "real" forecast, i.e., the first forecast that uses (2.4), is y_2.

Using (2.4), we can write

$$y_{t-1} = \alpha D_{t-2} + (1 - \alpha) y_{t-2},$$

so

$$y_t = \alpha D_{t-1} + \alpha(1 - \alpha) D_{t-2} + (1 - \alpha)^2 y_{t-2}.$$

We can continue the substitution in this way and eventually obtain

$$y_t = \sum_{i=0}^{\infty} \alpha(1 - \alpha)^i D_{t-i-1} = \sum_{i=0}^{\infty} \alpha_i D_{t-i-1},$$

where $\alpha_i = \alpha(1 - \alpha)^i$. The single exponential smoothing forecast includes all past observations, but since $\alpha_i < \alpha_j$ for $i > j$, the weights are decreasing as we move backward in time, as illustrated in Figure 2.1. Moreover,

$$\sum_{i=0}^{\infty} \alpha_i = \sum_{i=0}^{\infty} \alpha(1 - \alpha)^i = 1$$

by (C.50) in Appendix C. These weights can be approximated with an exponential function $f(i) = \alpha e^{-\alpha i}$. This is why this method is called exponential smoothing.

☐ **EXAMPLE 2.2**

Suppose that the book store manager from Example 2.1 wishes to use exponential smoothing to forecast next month's demand for *An Inventory Story*. Using $\alpha = 0.2$, we first initialize $y_1 = D_1$, and then obtain

$$y_2 = 0.2 D_1 + 0.8 y_1 = 10.61$$

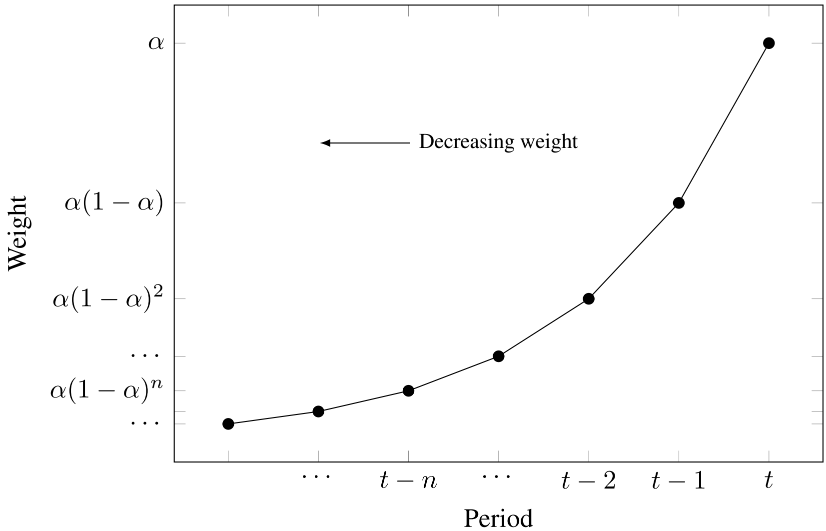


Figure 2.1 Weight distribution for single exponential smoothing.

$$y_3 = 0.2D_2 + 0.8y_2 = 10.89.$$
$$y_4 = 0.2D_3 + 0.8y_3 = 10.67$$

Continuing in this manner, we eventually get

$$y_{13} = 0.2D_{12} + 0.8y_{12} = 9.98.$$

□

2.2.2.2 Double Exponential Smoothing Double exponential smoothing can be used to forecast demands with a linear trend. Such demands can be modeled as follows:

$$D_t = I + tS + \epsilon_t, \tag{2.5}$$

where I is the base demand, S is the slope of the trend in the demand, and ϵ_t is an error term. The forecast for the demand in period t is the sum of two separate estimates from period $t-1$: one of the *base signal* (the value of the demand process) and one of the *slope*. That is,

$$y_t = I_{t-1} + S_{t-1}, \tag{2.6}$$

where I_{t-1} is the estimate of the base signal and S_{t-1} is the estimate of the slope, both made in period $t-1$. I_{t-1} represents our estimate of where the demand process fell in period $t-1$; in period t, the process will be S_{t-1} units greater. The estimates of the base signal and slope are calculated as follows:

$$I_t = \alpha D_t + (1-\alpha)(I_{t-1} + S_{t-1}) \tag{2.7}$$
$$S_t = \beta(I_t - I_{t-1}) + (1-\beta)S_{t-1}, \tag{2.8}$$

where α is the smoothing constant and β is the trend constant. Equation (2.7) is similar to (2.4) for single exponential smoothing in the sense that α is the weight placed on the most recent actual demand D_t and $1-\alpha$ is the weight on the previous forecast. Equation (2.8) can be explained similarly: It places a weight of β on the most recent estimate of the slope (obtained by taking the difference between the two most recent base signals) and a weight of $1-\beta$ on the previous estimate. Note that, if the trend is downward-sloping, then S_t will (usually) be negative.

As with single exponential smoothing, we need a way to initialize the process. This time, we need two historical demand observations to initialize the forecasts, and we typically set $I_1 = D_1$ and $S_1 = D_2 - D_1$ (then $y_2 = I_1 + S_1 = D_2$). The first "real" forecast (using (2.7)–(2.8) to get values for (2.6)) is y_3.

This particular version of double exponential smoothing is also known as *Holt's method* (Holt 1957).

☐ **EXAMPLE 2.3**

Suppose that the bookstore manager from Example 2.1 now turns her attention to another book (*The TSP Mystery*), with a different set of historical demand data, as presented in Table 2.1. In contrast to the stable demand of *An Inventory Story*, *The TSP Mystery*'s monthly demand data exhibits an increasing trend. Therefore, moving averages and single exponential smoothing may not accurately predict the demand of *The TSP Mystery* in the next month. For example, if we use a simple moving average of order $N = 5$, we get $y_{13} = 29.80$, which is much smaller than the demands in months 11 and 12. This may not be a good forecast, as we expect the demand in month 13 to continue to increase over time.

Instead, we will use double exponential smoothing for *The TSP Mystery*, with $\alpha = \beta = 0.2$. We initialize $I_1 = D_1 = 12.61$ and $S_1 = D_2 - D_1 = 3.40$. Then we have

$$
\begin{aligned}
y_2 &= I_1 + S_1 = 16.01 \\
I_2 &= 0.2D_2 + 0.8(I_1 + S_1) = 16.01 \\
S_2 &= 0.2(I_2 - I_1) + 0.8S_1 = 3.40 \\
y_3 &= I_2 + S_2 = 19.41 \\
I_3 &= 0.2D_3 + 0.8(I_2 + S_2) = 18.68 \\
S_3 &= 0.2(I_3 - I_2) + 0.8S_2 = 3.25.
\end{aligned}
$$

We continue this process and finally obtain

$$
\begin{aligned}
I_{12} &= 0.2D_{12} + 0.8(I_{11} + S_{11}) = 35.65 \\
S_{12} &= 0.2(I_{12} - I_{11}) + 0.8S_{11} = 1.94.
\end{aligned}
$$

So the forecast from double exponential smoothing is $y_{13} = I_{12} + S_{12} = 37.59$, which coincides with the increasing trend. ☐

2.2.2.3 Triple Exponential Smoothing Triple exponential smoothing can be used to forecast demands that exhibit both trend and seasonality. Seasonality means that the demand series has a pattern that repeats every N periods for some fixed N. N consecutive periods are called a *season*. (If the demand pattern repeats every year, for example, then a

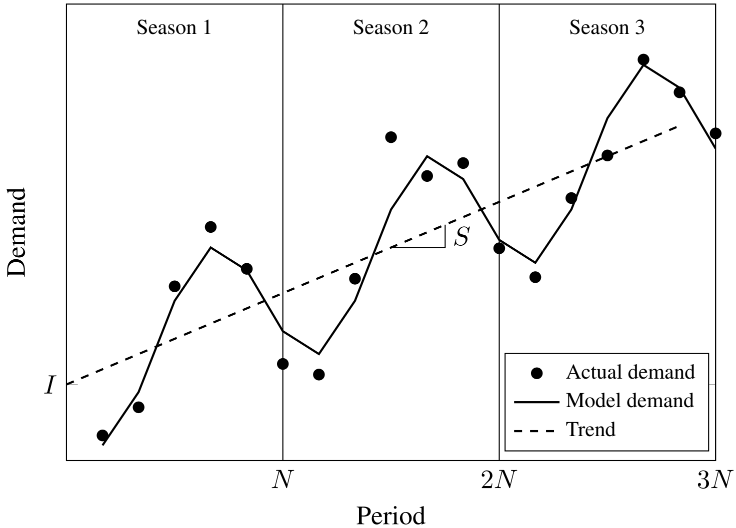


Figure 2.2 Random demands with trend and seasonality.

season is one year. This is different from the common usage of the word "season," which would refer to a portion of the year.)

To model the seasonality, we use a parameter c_t, $1 \le t \le N$, to represent the ratio between the average demand in period t and the overall average. (Thus, $\sum c_t = N$.) For example, if $c_6 = 0.88$, then on average, the demand in period 6 is 12% below the overall average demand. The c_t are called *seasonal factors*. We assume that the seasonal factors are unknown but that they are the same every season. The demand process can be modeled as follows:

$$D_t = (I + tS)c_t + \epsilon_t, \tag{2.9}$$

where I is the value of base signal at time 0, S is the true slope, and ϵ_t is a random error term. (See Figure 2.2.)

The forecast for period t is given by

$$y_t = (I_{t-1} + S_{t-1})c_{t-N}, \tag{2.10}$$

where I_{t-1} and S_{t-1} are the estimates of the base signal and slope in period $t-1$ and c_{t-N} is the estimate of the seasonal factor one season ago.

The idea behind smoothing with trend and seasonality is basically to "de-trend" and "de-seasonalize" the time series by separating the base signal from the trend and seasonality effects. The method uses three smoothing parameters, α, β, and γ, in estimating the base signal, the trend, and the seasonality, respectively:

$$I_t = \alpha \frac{D_t}{c_{t-N}} + (1-\alpha)(I_{t-1} + S_{t-1}) \tag{2.11}$$

$$S_t = \beta(I_t - I_{t-1}) + (1-\beta)S_{t-1} \tag{2.12}$$

$$c_t = \gamma \frac{D_t}{I_t} + (1-\gamma)c_{t-N}. \tag{2.13}$$

Equations (2.11) and (2.12) are very similar to (2.7) and (2.8) for double exponential smoothing, except that (2.11) uses the deseasonalized demand observation, D_t/c_{t-N}, instead of D_t, to average it with the current forecast. In (2.13), I_t is our estimate of the base signal, so D_t/I_t is our estimate of c_t based on the most recent demand. This is averaged with our previous estimate of c_t (made N periods ago) using weighting factor γ.

Initializing triple exponential smoothing is a bit trickier than for single or double exponential smoothing. To do so, we usually need at least two entire seasons' worth of data ($2N$ periods), which will be used for the initialization phase. One common method is to initialize the slope as

$$S_{2N} = \frac{1}{N}\left(\frac{D_{N+1}-D_1}{N} + \frac{D_{N+2}-D_2}{N} + \cdots + \frac{D_{2N}-D_N}{N}\right). \tag{2.14}$$

In other words, we take the per-period increase in demand between periods 1 and $N+1$, and the per-period increase between periods 2 and $N+2$, and so on; and then we take the average over those N values. To initialize the seasonal factors c_t, we estimate the seasonal factor for each period in the first two seasons, and then average them over those two seasons to obtain the initial seasonal factors:

$$c_{N+t} = \frac{1}{2}\left(\frac{D_t}{\sum_{j=1}^{N} D_j/N} + \frac{D_{N+t}}{\sum_{j=1}^{N} D_{N+j}/N}\right) \tag{2.15}$$

for $t = 1, \ldots, N$. Each denominator is the average demand in one season of the available data, so the fractions in the parentheses estimate the seasonal factor for the tth period in each season. The right-hand side as a whole averages these estimates over the two seasons. Finally, we estimate the base signal as $I_{2N} = D_{2N}/c_{2N}$. The first "real" forecast is y_{2N+1}.

This method is also sometimes known as *Winters's method* or the *Holt–Winters method* (Winters 1960).

☐ **EXAMPLE 2.4**

The book store described in Example 2.1 also sells CDs. The total monthly demand of all CDs in the last year is given in Table 2.1. Note that in addition to the increasing trend, the monthly demand has a seasonal pattern with seasons of one quarter: the demand in the first and third months of a quarter is about half of that in the second month of the same quarter. This observation motivates us to use triple exponential smoothing for demand forecasting.

Since the observed pattern repeats quarterly, i.e., every 3 months, we choose $N = 3$. To initialize the seasonal factors, we extract the average over the first two quarters:

$$c_4 = \frac{1}{2}\left(\frac{D_1}{(D_1+D_2+D_3)/3} + \frac{D_4}{(D_4+D_5+D_6)/3}\right) = 0.71$$
$$c_5 = \frac{1}{2}\left(\frac{D_2}{(D_1+D_2+D_3)/3} + \frac{D_5}{(D_4+D_5+D_6)/3}\right) = 1.51$$
$$c_6 = \frac{1}{2}\left(\frac{D_3}{(D_1+D_2+D_3)/3} + \frac{D_6}{(D_4+D_5+D_6)/3}\right) = 0.79.$$

The base signal and slope are initialized with the first two quarters as

$$I_6 = D_6/c_6 = 21.36$$

$$S_6 = \frac{1}{3}\left(\frac{D_4 - D_1}{3} + \frac{D_5 - D_2}{3} + \frac{D_6 - D_3}{3}\right) = 1.85.$$

Then, we forecast D_7 and update the signals and seasonality with $\alpha = \beta = \gamma = 0.2$ as follows:

$$\begin{aligned}
y_7 &= (I_6 + S_6)c_4 = 16.39 \\
I_7 &= 0.2\frac{D_7}{c_4} + 0.8(I_6 + S_6) = 23.91 \\
S_7 &= 0.2 \times (I_7 - I_6) + 0.8S_6 = 1.99 \\
c_7 &= 0.2 \times \frac{D_7}{I_7} + 0.8c_4 = 0.72.
\end{aligned}$$

Repeating this procedure for the subsequent periods, we ultimately obtain the final results:

$$\begin{aligned}
I_{12} &= 33.09 \\
S_{12} &= 1.86 \\
c_{12} &= 0.75 \quad c_{11} = 1.51 \quad c_{10} = 0.74 \\
y_{13} &= (I_{12} + S_{12})c_{10} = 25.90.
\end{aligned}$$

So, our forecast for the demand in month 13 is 25.90. □

2.2.3 Linear Regression

Historical data can also be used to forecast demands by determining a cause–effect relationship between some independent variables and the demand. For instance, the demand for sales of a brand of laptop computer may heavily depend on the sales price and the features. A regression model can be developed which describes this relationship. The model can then be used to forecast the demand for laptops with a given price and a given set of features.

In linear regression, the model specification assumes that the dependent variable, Y, is a linear combination of the independent variables. For example, in *simple linear regression*, there is one independent variable, X, and two parameters, β_0 and β_1:

$$Y = \beta_0 + \beta_1 X. \tag{2.16}$$

Here, X and Y are random variables. For any given pair of observed variables x and y, we have

$$y = \beta_0 + \beta_1 x + \epsilon, \tag{2.17}$$

where ϵ is a random error term. The objective of regression analysis is to estimate the parameters β_0 and β_1.

To build a regression model, we need historical data points—observations of both the independent variable(s) and the dependent variable. Let $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ be n paired data observations for a simple linear regression model. The goal is to find values of β_0 and β_1 so that the line defined by (2.16) gives the best fit of the data. In particular, β_0 and β_1 are chosen to minimize the sum of the squared residuals, where the residual for data point i is defined as the difference between the observed value of y_i and the predicted value of y_i obtained by substituting $X = x_i$ in (2.16). That is, we want to solve

$$\underset{\beta_0,\beta_1}{\text{minimize}} \sum_{i=1}^{n} \hat{e}_i^2 = \underset{\beta_0,\beta_1}{\text{minimize}} \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_i))^2 , \tag{2.18}$$

where $\hat{e}_i$ is the residual for data point i. The optimal values of β_0 and β_1 are given by

$$\hat{\beta}_1 = r_{xy}\frac{s_y}{s_x} \tag{2.19}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1\bar{x}, \tag{2.20}$$

where $\bar{x}$ and $\bar{y}$ are the sample means of the x_i and y_i, respectively; r_{xy} is the sample correlation coefficient between x and y; and s_x and s_y are the sample standard deviations of x and y, respectively (see, e.g., Tamhane and Dunlop (1999)).

If the demands exhibit a linear trend over time, then we can use regression analysis to forecast the demand using the time period itself (rather than, say, price or features) as the independent variable. In this case, it can be shown (see, e.g., Nahmias (2005, Appendix 2-B)) that the optimal values of β_0 and β_1 are given by:

$$\hat{\beta}_1 = \frac{A_{xy}}{A_{xx}} \tag{2.21}$$

$$\hat{\beta}_0 = \frac{1}{n}\sum_{i=1}^{n} D_i - \frac{\beta_1(n+1)}{2}, \tag{2.22}$$

where $D_1, \dots, D_n$ are the observed demands and

$$A_{xy} = n\sum_{i=1}^{n} iD_i - \frac{n(n+1)}{2}\sum_{i=1}^{n} D_i \tag{2.23}$$

$$A_{xx} = \frac{n^2(n+1)(2n+1)}{6} - \frac{n^2(n+1)^2}{4}. \tag{2.24}$$

According to the comparison by Carbonneau et al. (2008), linear regression often achieves better performance than moving average and trend methods.

☐ **EXAMPLE 2.5**

Return to the sales data for *The TSP Mystery* in Table 2.1. Rather than using double exponential smoothing to forecast the demand for period 13, as we did in Example 2.3, we can instead use linear regression. Using either (2.19)–(2.20) or (2.21)–(2.22), we get $\hat{\beta}_0 = 10.88$ and $\hat{\beta}_1 = 1.89$. Therefore, the forecast for the demand in period 13 is

$$10.88 + 13 \cdot 1.89 = 35.46.$$

The observed data and the best-fit line are plotted in Figure 2.3.

☐

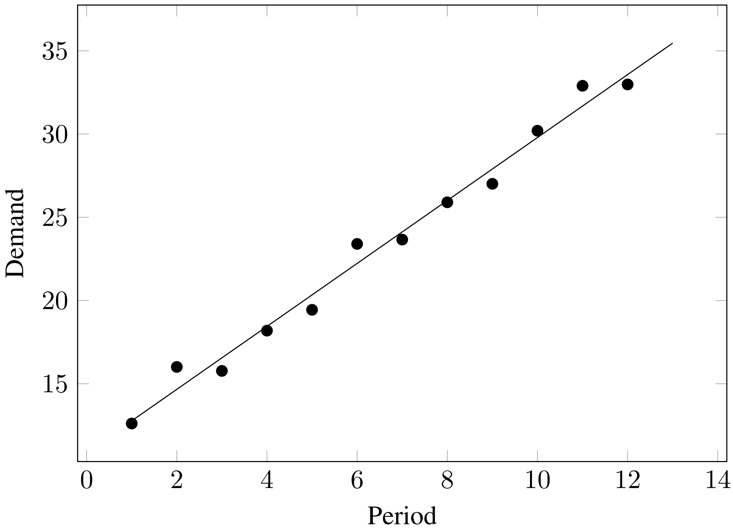


Figure 2.3 Observed demands for *The TSP Mystery* and best-fit line for Example 2.5.

2.3 FORECAST ACCURACY

2.3.1 MAD, MSE, and MAPE

At some point after a forecast is computed, the actual demand is observed, providing us with an opportunity to evaluate the quality of the forecast. The most basic measure of the forecast accuracy is the *forecast error*, denoted e_t, which is defined as the difference between the forecast for period t and the actual demand for that period:

$$e_t = y_t - D_t, \tag{2.25}$$

where y_t is a forecast obtained using any method and D_t is the actual observed demand.

Since the forecast and the demand are random variables, so is the forecast error; let μ_e and σ_e^2 denote its mean and variance, respectively. If the mean of the forecast error, μ_e, equals 0, we say the forecasting method is *unbiased*: It does not produce forecasts that are systematically either too low or too high. However, even an unbiased forecasting method can still be very inaccurate. One way to measure the accuracy is using the variance of the forecast error, σ_e^2. To compute μ_e or σ_e^2, however, we need to know the probabilistic process that underlies both the demands and the forecasts. Typically, therefore, we use performance measures based on sample quantities rather than population quantities.

Two of the most common such measures are the *mean absolute deviation (MAD)* and the *mean squared error (MSE)*, defined as follows:

$$\text{MAD} = \frac{1}{n}\sum_{t=1}^{n} |e_t| \tag{2.26}$$

$$\text{MSE} = \frac{1}{n}\sum_{t=1}^{n} e_t^2. \tag{2.27}$$

MSE is identical to the sample variance of the random forecast error e_t except for the denominator of the coefficient. MAD is sometimes preferred to MSE in real applications because it avoids the calculation of squaring, though modern spreadsheet and statistics packages can compute either performance measure easily. When the forecast errors are normally distributed, their standard deviation is often estimated as

$$\sigma_e \approx 1.25\text{MAD}. \tag{2.28}$$

This is useful when σ_e is required (e.g., for inventory optimization models—see Section 4.3.2.7), since, as previously noted, we do not typically know σ_e directly.

Note that both MAD and MSE are dependent on the magnitude of the values of demand; if we express the demands in different units (e.g., tons vs. pounds), the performance measures will change. By comparison, the *mean absolute percentage error (MAPE)* is independent of the magnitude of the demand values:

$$\text{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{e_t}{D_t}\right| \times 100. \tag{2.29}$$

☐ **EXAMPLE 2.6**

Table 2.2 gives the hypothetical actual demands for periods 13–24 for *An Inventory Story* from Example 2.1. It also gives the moving average forecasts for these periods (using $N = 5$), the single exponential smoothing forecasts for these periods (using $\alpha = 0.2$), and the corresponding forecast errors. Finally, at the end of the table are the performance measures (MAD, MSE, and MAPE) for each of the forecasting methods. In this case, the moving average has slightly smaller values of the performance measures and is therefore slightly more accurate. ☐

2.3.2 Forecast Errors for Moving Average and Exponential Smoothing

Assume that the demand is generated by the process

$$D_t = \mu + \epsilon_t, \tag{2.30}$$

where $\epsilon_t \sim N(0, \sigma^2)$. Since the demand process is stationary, either moving average or exponential smoothing is an appropriate forecasting method.

In a moving average of order N, the forecast y_t is given by (2.2). It follows that

$$\mu_e = \mathbb{E}[y_t - D_t] = \frac{1}{N}\sum_{i=t-N}^{t-1}\mathbb{E}[D_i] - \mathbb{E}[D_t] = \frac{1}{N}N\mu - \mu = 0.$$

Therefore, moving-average forecasts are unbiased when the demand is stationary.

We can also derive the variance of the forecast error, which can be expressed as

$$\begin{aligned}\sigma_e = \sqrt{\text{Var}[y_t - D_t]} &= \sqrt{\text{Var}[y_t] + \text{Var}[D_t]} \\ &= \sqrt{\frac{1}{N^2}\sum_{i=t-N}^{t-1}\text{Var}[D_i] + \text{Var}[D_t]}\end{aligned}$$

Table 2.2 Demands (D_t), forecasts (y_t), and forecast errors (e_t) for *An Inventory Story*, periods 13–24, for Example 2.6.

		Moving Average		Exponential Smoothing	
t	D_t	y_t	e_t	y_t	e_t
13	10.98	9.80	−1.18	9.98	−1.00
14	12.07	10.01	−2.06	10.18	−1.89
15	11.45	10.63	−0.82	10.56	−0.89
16	9.39	10.88	1.49	10.74	1.35
17	10.59	10.57	−0.02	10.47	−0.12
18	8.43	10.90	2.47	10.49	2.06
19	11.78	10.39	−1.39	10.08	−1.70
20	7.71	10.33	2.62	10.42	2.71
21	7.86	9.58	1.72	9.88	2.02
22	8.38	9.27	0.89	9.47	1.09
23	4.11	8.83	4.72	9.26	5.15
24	12.88	7.97	−4.91	8.23	−4.65
MAD			2.02		2.05
MSE			6.13		6.26
MAPE			25.97		26.85

$$
\begin{aligned}
&=\sqrt{\frac{1}{N^2}N\sigma^2+\sigma^2} \\
&=\sigma\sqrt{\frac{1+N}{N}}.
\end{aligned}
$$

Note that the second equality uses the fact that the forecast and demand in period t are statistically independent.

If forecasts are instead performed using exponential smoothing, one can show (see Problem 2.12) that

$$\mu_e = 0 \tag{2.31}$$

$$\sigma_e = \sigma\sqrt{\frac{2}{2-\alpha}}. \tag{2.32}$$

2.4 MACHINE LEARNING IN DEMAND FORECASTING

2.4.1 Introduction

We are in the age of big data. The huge volume of data generated every day, the high velocity of data creation, and the large variety of sources all make today's business information environment different than it was only a decade ago. Using data intelligently is key to business decision-making. A 2012 *Harvard Business Review* article notes: "Data-driven decisions are better decisions—it's as simple as that. Using big data enables managers to decide on the basis of evidence rather than intuition. For that reason it has the potential to revolutionize management" (McAfee and Brynjolfsson 2012).

Fortunately, many businesses have access to large volumes of historical demand data that can help when forecasting future demands. In this section, we introduce some of the main machine learning techniques for demand forecasting. Compared with classical forecasting methods such as the time series methods discussed in Section 2.2, machine learning models often significantly increase prediction accuracy.

2.4.2 Machine Learning

In general, *machine learning* (ML) refers to a set of algorithms that can learn from and make predictions about data. These algorithms take data as inputs and generate predictions or decisions as outputs. Machine learning is closely related to *statistical learning*, which refers to a set of tools for modeling and understanding complex data sets (James et al. 2013). Machine learning and statistical learning have developed rapidly in recent years. Both techniques fall into the overall field of *data science*, which covers a wider range of topics, including database design and data visualization techniques.

One category of ML algorithms is called *supervised learning*, in which the historical data contain both inputs and outputs, and the learning algorithm learns to predict an output for a given set of inputs. For example, we might have historical data that contains the outdoor temperature and the number of glasses of lemonade that were sold on each day. The learning algorithm tries to infer the relationship between the two, so that for a given temperature, it can predict the number of glasses of lemonade that will be sold. Regression is a simple example. In contrast, *unsupervised learning* explores relationships and structures within the data without any known "ground truth" labels or outputs. For example, if we wish to partition consumers into market segments, we might use a clustering algorithm, which is a type of unsupervised learning. (See Friedman et al. (2001) or James et al. (2013) for further discussion of this dichotomy.) Demand forecasting falls into the category of supervised learning since we need to predict future demands (outputs) using historical demand data and other market information (inputs).

Common supervised learning methods include linear regression (and its nonlinear extensions), kernel methods, tree-based models, support vector machines (SVMs), and neural networks. Graphical models involving hidden Markov models (or, in their simplest form, mixture models) and Markov random fields also receive considerable attention. In the following subsections, we discuss the learning methods that are most commonly applied to demand forecasting.

2.4.2.1 Linear Regression Linear regression is a very simple supervised learning method. It assumes that the output Y is linear in the inputs $X_1, X_2, \ldots, X_p$, where p is the number of distinct input variables (also called *predictors* or *features*):

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p. \tag{2.33}$$

For particular values of the inputs and outputs, we have

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon, \tag{2.34}$$

where ϵ is a random error term. The β_js are coefficients that need to be estimated from data. If $p = 1$, then we have *simple linear regression*, which we discuss in Section 2.2.3.

Table 2.3 Snippet of historical data on demand for baseball jerseys for Examples 2.7–2.9.

Batting Avg.	Games Won	Years in Majors	Demand (Cases)
0.274	68	1	14.3
0.332	150	11	28.7
0.262	79	12	17.6
0.396	127	8	26.0
0.262	156	4	27.1
0.280	142	7	26.0
0.112	75	10	14.7
0.429	82	0	19.2
0.259	88	7	18.1
0.302	95	6	19.4

(In Section 2.2.3, we focused on the use of time as the independent variable in order to predict demands as a function of time. Here, our independent variables can be any feature.)

The most common way to obtain the β_js is *least squares*, which seeks to find the minimizer of the sum of the squares of the residuals. (Recall from Section 2.2.3 that the residual for data point i is the difference between the observed and predicted values of y_i.) The derived estimated coefficients are denoted $\hat{\beta}_j$. Then we can make predictions on new inputs by using

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_p x_p, \tag{2.35}$$

where $\hat{y}$ is our predicted value for the output, given the observed values $\{x_1, x_2, \ldots, x_p\}$ of the inputs.

☐ **EXAMPLE 2.7**

A sports apparel retailer sells jerseys (T-shirts) with the names of major league baseball players stitched onto the back. The retailer believes that the demand for a given player's jersey depends on his batting average last year, the number of games his team won last year, and the number of years the player has been playing in the major leagues. Therefore, the retailer keeps careful records of these statistics, as well as the demand for jerseys, for each player. Last year's records for 300 players can be found in the file `jerseys.xlsx`, the first few rows of which are reproduced in Table 2.3. Demands are expressed in cases sold last year. (In baseball, batting averages are expressed as decimals between 0 and 1.)

The retailer wishes to predict the demand for this year's jerseys using the historical data. Let X_1 represent batting average, X_2 represent games won, and X_3 represent years in majors. Using regression, we find that

$$\begin{aligned}
\hat{\beta}_0 &= -0.0651 \\
\hat{\beta}_1 &= 18.0430 \\
\hat{\beta}_2 &= 0.1403 \\
\hat{\beta}_3 &= 0.1831.
\end{aligned}$$

For example, if Roy Hobbs had a 0.292 average last year, his team won 95 games, and he has been in the major leagues for 4 years, the demand for his jersey this year can be predicted as

$$\hat{y} = -0.0651 + 18.0430 \times 0.292 + 0.1403 \times 95 + 0.1831 \times 4 = 19.2644.$$

□

Although the linear regression model assumes a linear relationship between the output and the inputs, we can model nonlinear relationships by introducing basis functions and splines. When the number of predictors is large, we can utilize shrinkage methods such as least absolute shrinkage and selection operator (LASSO) and ridge regression. In general, linear regression is a simple but strong learning method.

2.4.2.2 Tree-based models Tree-based models use decision trees to make predictions for a given set of inputs. They can be applied both to regression problems (in which the outputs are continuous) and to classification problems (in which the outputs are categorical). The trees used for these two types of problems are referred to as *regression trees* and *classification trees*, respectively. In demand forecasting, regression trees have received more attention because of their simplicity and interpretability.

A regression tree divides the space of input variables, i.e., the set of possible values of $X_1, X_2, \ldots, X_p$, into distinct and nonoverlapping regions and assigns a single output, c_k, to each region k. If a given input $x_1, x_2, \ldots, x_p$ falls into region k, then the demand forecast y for that input is equal to c_k. The c_k values are determined simply by averaging the observations in the historical data that fall into that region.

The goal is to choose the partition strategy that minimizes the sum of squares of the residuals, similar to linear regression. However, in practice, the number of possible partitions may be too large to enumerate. Therefore, it is common to use a binary splitting method called *recursive partitioning*, which generates two regions from the original region at each iteration. For the purposes of prediction, the size of the tree is limited by a pruning process. A single tree may not perform well due to high variance of the forecast, so researchers have developed methods that combine several trees to enhance the prediction performance. These include *random forests*, *bagging*, and *boosting*.

Tree-based models are used widely in demand forecasting for many industries. For example, Ferreira et al. (2015) apply regression trees with bagging to predict the demand of new styles for an online retailer. They show that tree-based models outperform linear regression and some nonlinear regression models consistently. Ali et al. (2009) develop regression trees to predict stock-keeping unit (SKU) sales for a European grocery retailer. They incorporate information about current promotions when constructing regression trees and show that regression trees provide better accuracy than linear regression and SVMs.

□ **EXAMPLE 2.8**

Return to the baseball-jersey data set from Example 2.7. Figure 2.4 shows one possible regression tree for this data set. For example, we would predict a demand of 23.5 cases for Roy Hobbs (who has a 0.292 batting average, has been in the major leagues for 4 years, and is on a team that has won 95 games).

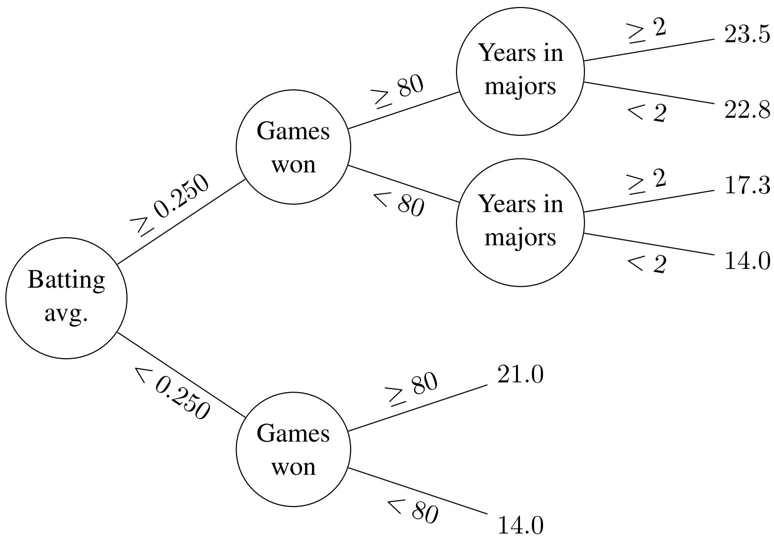


Figure 2.4 Regression tree for baseball jerseys for Example 2.8.

Of course, there are many possible regression trees for this data set, and the figure gives only one example. Better ones can be found using the recursive partitioning method.

□

2.4.2.3 Support vector machines SVMs are designed to partition the space of input variables into two regions, i.e., to make a binary prediction about a given output based on which region a given input vector falls into. The partition is accomplished by finding a separating hyperplane. In particular, assuming that the training data set is linearly separable, the optimal separating hyperplane is found by solving the following optimization problem:

$$\underset{\beta_0,\beta}{\text{minimize}} \quad ||\beta||_2^2 \tag{2.36}$$

$$\text{subject to} \quad y^i(\mathbf{x}^i \cdot \beta + \beta_0) \geq 1 \quad \forall i = 1, 2, \ldots, N, \tag{2.37}$$

where N is the number of observations, y^i is the binary output ($y^i \in \{0, 1\}$) for observation i, $\mathbf{x}^i \in \mathbb{R}^p$ is the vector of input variables for observation i, and $\cdot$ denotes dot product. This is also called a *maximum margin classifier*, where the *margin* is defined as $\frac{1}{||\beta||_2}$. The optimal values of the vector $\boldsymbol{\beta} \in \mathbb{R}^p$ and the scalar β_0 characterize the separating hyperplane. For a given input vector $x_1, \ldots, x_p$, we predict an output value of 1 if $\mathbf{x}^i \cdot \boldsymbol{\beta} + \beta_0 > 0$ and a value of 0 otherwise.

For example, suppose we wish to predict which customers will purchase a product based on their age, income, and money spent at the store in the past year. We code each customer in the historical data with a 1 or 0 depending on whether they purchased the product, then solve (2.36)–(2.37) to find the hyperplane that does the best job of separating the 1s from the 0s. For each new customer, we simply calculate $\mathbf{x}^i \cdot \boldsymbol{\beta} + \beta_0$ and make a prediction accordingly.

SVMs can be generalized to allow nonlinearities by mapping the input space into a high-dimensional space using *kernel functions*. In essence, this allows the region to be partitioned using a surface that is not linear, i.e., is not a hyperplane. Popular choices of kernel functions include polynomials and radial basis functions (RBFs).

Since SVMs can be used to make binary predictions, they can be used to predict whether a given customer will purchase a product. They can also be used to forecast the demand as a quantity using *support vector regression* (SVR), an adaptation of the SVM approach to regression problems using kernel functions. SVR is among the best machine learning methods for supply chain demand forecasting (Carbonneau et al. 2008).

☐ **EXAMPLE 2.9**

For the baseball-jersey data set from Example 2.7, let us first use SVM to predict whether the demand for a given player's jerseys will be greater than or equal to 25 cases this year. We can label the historical data by assigning $y^i = 1$ to players whose jerseys had a demand greater than or equal to 25 and $y^i = 0$ for those who did not. Solving the SVM optimization problem[1] results in the solution $\boldsymbol{\beta} = (4.5879, 0.0745, 0.1154)$ and $\beta_0 = -12.1620$. In other words, if

$$-12.1620 + (4.5879, 0.0745, 0.1154) \cdot (x_1, x_2, x_3) > 0,$$

then we predict that the demand will be greater than or equal to 25. For Roy Hobbs, who has an input vector of $\mathbf{x}^i = (0.292, 95, 4)$, we have

$$-12.1620 + (4.5879, 0.0745, 0.1154) \cdot (0.292, 95, 4) = -3.2832,$$

so we predict that Roy will *not* sell more than 25 cases of jerseys this year.

Next, we can use an SVR model to predict the demand for Roy Hobbs jerseys explicitly. Using MATLAB's `fitrsvm` function, we obtain SVR coefficients of $\boldsymbol{\beta} = (13.8451, 0.1387, 0.1932)$ and $\beta_0 = 1.1436$. Therefore, we can predict the demand for Roy Hobbs jerseys as

$$1.1436 + (13.8451, 0.1387, 0.1932) \cdot (0.292, 95, 4) = 19.1357 \text{ cases.}$$

(Note that the SVM and SVR optimization problems are nonconvex and typically have multiple optima. Your results might differ if you use a different implementation to solve the same problem.) ☐

2.4.2.4 Neural Networks A neural network consists of several *nodes*, also called *neurons*, arranged into *layers*. The first layer of nodes represents the inputs (the X_i values); the last layer represents the outputs (the Y value); and one or more layers in between, called *hidden layers*, process the information from the input layer and perform the actual computation of the network. (See Figure 2.5.) Neural networks have been used extensively for classification problems such as image and speech processing, where the

[1] We did not use (2.36)–(2.37), but rather a modified formulation, since the training data set in this example is not linearly separable. We used MATLAB's `fitcsvm` function to do the optimization.

goal is to determine what sort of physical or linguistic object the inputs represent. But neural networks can and have been successfully applied to regression-type problems such as demand forecasting.

The central idea behind neural networks is that in each layer (except the first), we extract linear combinations of the inputs from the previous layer as derived features, and then model the output as a nonlinear function of these features. For example, in a typical network with a single hidden layer with M nodes, each hidden-layer node $m = 1, \ldots, M$ calculates the derived feature

$$Z_m = \sigma(\alpha_{0m} + \boldsymbol{\alpha}_m^T \mathbf{X}), \tag{2.38}$$

where $\mathbf{X}$ is the vector of inputs, α_{0m} is a scalar, $\boldsymbol{\alpha}_m$ is a vector with p elements (one per input feature), and $\sigma(\cdot)$ is a nonlinear function called the *activation function*. Note that the term inside the $\sigma(\cdot)$ is a linear combination of the inputs plus a constant. Typical activation functions include the sigmoid function and the ReLU function. The Z_m are also called *hidden units* since they are not directly observed. Once the hidden units are calculated by the hidden-layer nodes, the output Y is modeled as a function of the hidden units:

$$Y = g(Z_1, \ldots, Z_M), \tag{2.39}$$

where $g(\cdot)$ is a (possibly nonlinear) function.

The key challenge in fitting a neural network model is the determination of the weights α_{0m} and α_m. This is usually done using some sort of algorithm that modifies the weights as the network "learns" right and wrong answers. The most common such algorithm is known as *backpropagation*, which calculates gradients with respect to the weights; another method (such as *gradient descent*) is then used to update the weights. Determining these weights—sometimes referred to as training the network—can be computationally intensive. However, once the network is trained, generating an output value for a new set of inputs is extremely efficient. (For further details, see, e.g., Friedman et al. (2001).)

Some neural networks contain multiple hidden layers, not just one; this can improve the accuracy of the network's predictions but makes the network harder to train. Such *deep neural networks* have led to huge advances in machine learning, with great successes not only in classification and prediction problems such as image processing and demand forecasting, but also, when coupled with reinforcement learning (RL), in solving decision problems such as those in board games; one famous example is Google DeepMind's AlphaGo program, which beat the world-champion (human) Go player in 2016.

Carbonneau et al. (2008) test two different types of neural networks on demand forecasting and conclude that neural networks perform better than traditional methods. Venkatesh et al. (2014) combine neural networks with clustering to predict demand for cash at automatic teller machines (ATMs). They find that their model increases the prediction accuracy substantially.

2.5 DEMAND MODELING TECHNIQUES

As the pace of technology accelerates, companies are introducing new products faster and faster to stay competitive. There is a diffusion process associated with the demand for any new product, so companies need to plan the timing and quantity of new product releases

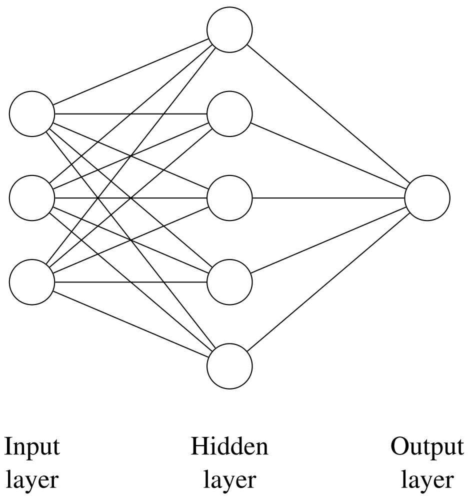


Figure 2.5 A simple neural network.

carefully to match supply and demand as closely as possible. To do so, they need to understand the life cycles and demand dynamics of their products.

One of the authors has worked with a high-tech company in China. The company was complaining about their very inaccurate demand forecasts, which led to excess inventory valued at approximately $25 million. The author was invited to give lectures on demand forecasting and inventory management. The first day's lecture focused on the classical time-series demand forecasting techniques discussed earlier in this chapter. The reaction from the company's forecasting team was lukewarm. They were already quite familiar with these techniques and had tried hard to make them work, unsuccessfully. It turns out that classical forecasting techniques did not work well with the company's highly variable, short-life-cycle products, so the firm introduced products at the wrong times in the wrong quantities. The forecasting team's reaction was quite different when the author discussed the Bass diffusion model, the leading-indicator method, and choice models, which are designed to account for short life cycles and other important factors. We discuss each of these methods in detail in the following sections. (As a postscript, the company reported more than a 50% increase in sales about one and a half years after they improved their forecasting techniques, partially due to the fact that money was being invested in a better mix of products.)

2.6 BASS DIFFUSION MODEL

The sales patterns of new products typically go through three phases: rapid growth, maturity, and decline. The *Bass diffusion model* (Bass 1969) is a well-known parametric approach for estimating the demand trajectory of a single new product over time. Bass's basic three-parameter model has proved to be very effective in delivering accurate forecasts and insights for a huge variety of new product introductions, regardless of pricing and advertising decisions. The model forecasts well even when limited or no historical data

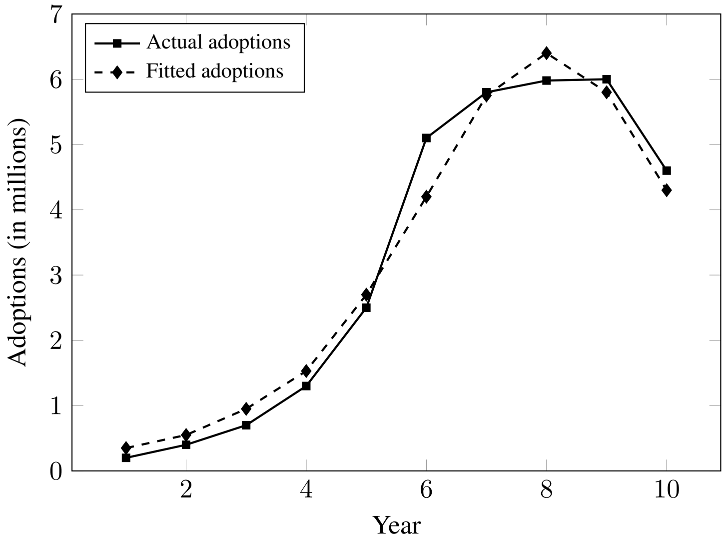


Figure 2.6 Color TVs in the 1960s: Forecasts from Bass model and actual demands. Reprinted by permission, Bass, Empirical generalizations and marketing science: A personal view, *Marketing Science*, 14(3), 1995, G6–G19. ©1995, the Institute for Operations Research and the Management Sciences (INFORMS), 7240 Parkway Drive, Suite 300, Hanover, MD 21076 USA.

are available. For example, Figure 2.6 depicts demand data (forecast and actual) for the introduction of color television sets in the 1960s.

The premise of the Bass model is that customers can be classified into *innovators* and *imitators*. Innovators (or *early adopters*) purchase a new product without regard to the decisions made by other individuals. Imitators, on the other hand, are influenced in the timing of their purchases by previous buyers through word-of-mouth communication. Refer to Figure 2.7 for an illustration. The number of innovators decreases over time, while the number of imitators purchasing the product first increases, and then decreases. The goal of the Bass model is to characterize this behavior in an effort to forecast the demand. It mathematically characterizes the word-of-mouth interaction between those who have adopted the innovation and those who have not yet adopted it. Moreover, it attempts to predict two important dimensions of a forecast: how many customers will eventually adopt the new product, and when they will adopt. Knowing the timing of adoptions is important as it can guide the firm to smartly utilize resources in marketing the new product. Our analysis of this model is based on that of Bass (1969).

2.6.1 The Model

The Bass model assumes that $P(t)$, the probability that a given buyer makes an initial purchase at time t given that she has not yet made a purchase, is a linear function of the number of previous buyers; that is,

$$P(t) = p + \frac{q}{m} D(t), \tag{2.40}$$

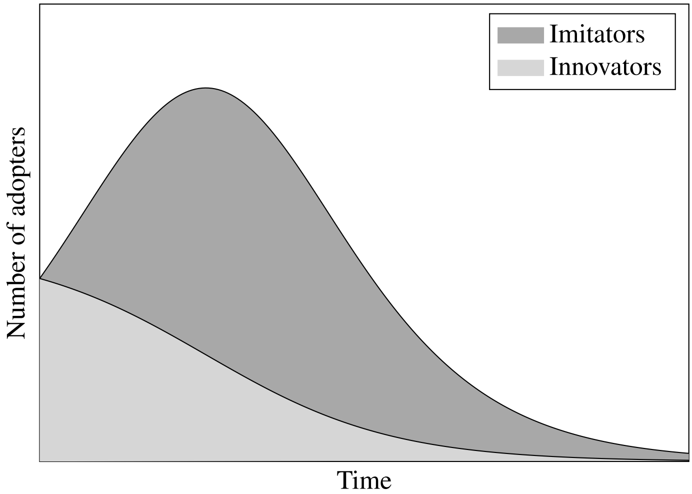


Figure 2.7 Bass diffusion curve.

where $D(t)$ is the cumulative demand by time t. Equation (2.40) suggests that two factors will influence the probability that a customer makes a purchase at time t. The first factor is the *coefficient of innovation*, denoted p, which is a constant, independent of how many other customers have adopted the innovation before time t. The second factor, $\frac{q}{m}D(t)$, measures the "contagion" effect between the innovators and the imitators and is proportional to the number of customers who have already adopted by time t. The parameters q and m represent the *coefficient of imitation* and the *market size*, respectively. We require $p < q$. In fact, usually $p \ll q$; for example, $p = 0.03$ and $q = 0.38$ have been reported as average values (Sultan et al. 1990).

We assume that the time index, t, is measured in years. Of course, any time unit is possible, but the values we report for p and q implicitly assume that t is measured in years.

Let $d(t)$ be the derivative of $D(t)$, i.e., the demand *rate* at time t. Using Bayes' rule, one can show that

$$P(t) = \frac{d(t)}{m - D(t)}. \tag{2.41}$$

(See Section 2.6.2 for a derivation of the analogous equation in the discrete-time model.) Combining (2.40) and (2.41), we have

$$d(t) = \left(p + \frac{q}{m}D(t)\right)(m - D(t)). \tag{2.42}$$

Our goal is to characterize $D(t)$ so that we can understand how the demand evolves over time. To a certain extent, (2.42) does this, but (2.42) is a differential equation; it expresses $D(t)$ in terms of its derivative. Our preference would be to have a closed-form expression for $D(t)$. Fortunately, this is possible:

Theorem 2.1

$$D(t) = m\frac{1 - e^{-(p+q)t}}{1 + \frac{q}{p}e^{-(p+q)t}} \tag{2.43}$$

$$d(t) = \frac{mp(p+q)^2 e^{-(p+q)t}}{\left(p + qe^{-(p+q)t}\right)^2} \tag{2.44}$$

Proof. Omitted. ■

As a corollary, one can determine the time at which the demand rate peaks, and the demand rate and cumulative demand at that point:

Corollary 2.2 *The peak demand occurs at time*

$$t^* = \frac{1}{p+q} \ln\left(\frac{q}{p}\right). \tag{2.45}$$

The demand rate and cumulative demand at time t^* are given by

$$d(t^*) = \frac{m(p+q)^2}{4q} \tag{2.46}$$

$$D(t^*) = \frac{m(q-p)}{2q}. \tag{2.47}$$

Proof. Omitted; see Problem 2.17. ■

If p is very small, then the demand growth occurs slowly, whereas if p and q are large, sales take off rapidly and fall off quickly after reaching their maximum. Note that the formulas in Corollary 2.2 are only well defined if $q > p$, which we previously assumed to be true. If, instead, $q < p$, then the innovation effects will dominate the imitation effects, and the peak demand will occur immediately upon the introduction of the product and will decline thereafter. In summary, by varying the values of p and q, we can represent many different patterns of demand diffusion.

☐ **EXAMPLE 2.10**

The bookstore manager from Example 2.3 now wishes to model the demand for a third book, *The Case of the Violated Constraint*, which is expected to be a best-seller but whose sales will taper off after their peak. The bookstore's marketing department has estimated that the sales of the book will follow a Bass diffusion process with parameters $p = 0.05$, $q = 0.3$, and $m = 2700$, which are calculated assuming that the time index is measured in weeks (not years).

At what time will the sales of *The Case of the Violated Constraint* reach their peak, and what will the demand rate be at that time? How many copies of the book will have been sold by that point? What will the demand rate be at week 20, and how many copies will have been sold by that point?

From Corollary 2.2, we have

$$t^* = \frac{1}{0.05 + 0.3} \ln\left(\frac{0.3}{0.05}\right) = 5.12,$$

so the peak occurs during week 5. Moreover,

$$d(t^*) = \frac{2700(0.05 + 0.3)^2}{4 \cdot 0.3} = 27.63$$

$$D(t^*) = \frac{2700(0.3 - 0.05)}{2 \cdot 0.3} = 1125.00$$

and, from (2.43)–(2.44),

$$D(20) = 2700 \frac{1 - e^{-(0.05+0.3)\cdot 20}}{1 + \frac{0.3}{0.05} e^{-(0.05+0.3)\cdot 20}} = 2682.86$$

$$d(20) = \frac{2700 \cdot 0.05(0.05 + 0.3)^2 e^{-(0.05+0.3)\cdot 20}}{\left(0.05 + 0.3 e^{-(0.05+0.3)\cdot 20}\right)^2} = 5.97.$$

Therefore, at the time of peak demand, the demand rate will be 27.63 books per week, and 1125 books will have been sold. At week 20, the demand rate will be 5.97 books per week, and 2682.86 (or 2683) books will have been sold. □

Seasonal influence factors can be incorporated into the Bass framework. Kurawarwala and Matsuo (1996) present a growth model to forecast demand for short-life-cycle products that is motivated by the Bass diffusion model. They use α_t to denote the seasonal influence parameter at time t, given as a function with a periodicity of 12 months. Their proposed seasonal growth model is characterized by the following differential equation:

$$d(t) = \left(p + \frac{q}{m} D(t)\right)(m - D(t))\alpha_t, \tag{2.48}$$

where $D(t)$ is the cumulative demand by time t ($D(0) \equiv 0$), $d(t)$ is its derivative, and m, p, and q are the scale and shape parameters, which are analogous to the parameters in the Bass diffusion model. This is identical to (2.42) except for the multiplier α_t.

Integrating (2.48), we get the cumulative demand $D(t)$ as follows:

$$D(t) = m \left[\frac{1 - e^{-(p+q)\int_0^t \alpha_\tau d\tau}}{1 + \frac{q}{p} e^{-(p+q)\int_0^t \alpha_\tau d\tau}} \right]. \tag{2.49}$$

When $\alpha_t = 1$ for all t, (2.49) reduces to (2.43) from Bass's original model.

2.6.2 Discrete-Time Version

A discrete-time version of the Bass model is available. In this case, d_t represents the demand in period t, and D_t represents the cumulative demand up to period t. Let P_t be the probability that a customer buys the product in period t given that she did not buy it in periods $1, \ldots, t-1$. Bayes' rule says that

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(B|A)\mathbb{P}(A)}{\mathbb{P}(B)}.$$

Here, let A represent "customer buys in t" and B represent "customer didn't buy in $1, \ldots, t-1$." Then

$$\mathbb{P}(A|B) = \frac{1 \cdot \frac{d_t}{m}}{1 - \frac{D_t}{m}} = \frac{d_t}{m - D_t}.$$

(Note the similarity to (2.41), which is for continuous time.) Then the discrete-time analogue of (2.42) is

$$d_t = \left(p + \frac{q}{m} D_{t-1}\right)(m - D_{t-1}), \tag{2.50}$$

where $D_0 \equiv 0$.

2.6.3 Parameter Estimation

The Bass model is heavily driven by the parameters m, p, and q. In this section, we briefly discuss how these parameters may be estimated.

If historical data are available, we can estimate the parameters p, q, and m by first finding the least-squares estimates of the parameters a, b, and c in the following linear regression model:

$$d_t = a + bD_{t-1} + cD_{t-1}^2 \qquad t = 2, 3, \ldots.$$

Note that this model uses the discrete-time version of the Bass model (in which we observe demands d_t and calculate cumulative demands D_t) since, in practice, we observe discrete demand quantities rather than a continuous demand function. After finding a, b, and c using standard regression analysis, the parameters of the Bass model can be determined as follows:

$$m = \frac{-b - \sqrt{b^2 - 4ac}}{2c} \tag{2.51}$$

$$p = \frac{a}{m} \tag{2.52}$$

$$q = -mc. \tag{2.53}$$

However, because the Bass model is typically used for new products, in most cases historical data are not available to estimate the parameters. Instead, m is typically estimated qualitatively, using judgment or intuition from management about the size of the market, market research, or the Delphi method. In some markets these estimates can be rather precise. For instance, the pharmaceutical industry is known for their accurate demand estimates, which derive from abundant data regarding the incidence of diseases and ailments (Lilien et al. 2007). The parameters p and q tend to be relatively consistent within a given industry, so these can often be estimated from the diffusion patterns of similar products. Lilien and Rangaswamy (1998) provide industry-specific data for a wide range of industries. (See Table 2.4 for some examples.)

2.6.4 Extensions

After more than half a century, the Bass model is still actively used in demand forecasting and production planning. Sultan et al. (1990), Mahajan et al. (1995), and Bass (2004) provide broad overviews of these applications. The original model has also been extended in a number of ways. Ho et al. (2002) provide a joint analysis of demand and sales dynamics when the supply is constrained, and thus the usual word-of-mouth effects are mitigated. Their analysis generalizes the Bass model to include backorders and lost sales and describes the diffusion dynamics when the firm actively makes supply-related decisions to influence the diffusion process. Savin and Terwiesch (2005) describe the demand dynamics of two new products competing for a limited target market, generalizing the innovation and imitation effects in Bass's original model to account for this competition. Schmidt and Druehl (2005) explore the influence of product improvements and cost reductions on the new-product diffusion process. Ke et al. (2013) consider the problem of extending a product line while accounting for both inventory (supply) and diffusion (demand). The model determines whether and when to introduce the line extension and the corresponding production quantities. Islam (2014) uses the Bass model (as well as experimental discrete

Table 2.4 Bass model parameters. Adapted with permission from Lilien and Rangaswamy, *Marketing Engineering: Computer-Assisted Marketing Analysis and Planning*, Addison-Wesley, with permission obtained from Pearson, 1998, p. 201.

Product	p	q
Cable TV	0.100	0.060
Camcorder	0.044	0.304
Cellular phone	0.008	0.421
CD player	0.157	0.000
Radio	0.027	0.435
Home PC	0.121	0.281
Hybrid corn	0.000	0.797
Tractor	0.000	0.234
Ultrasound	0.000	0.534
Dishwasher	0.000	0.179
Microwave	0.002	0.357
VCR	0.025	0.603

choice data—see Section 2.8) to predict household adoption of photovoltaic (PV) solar cells.

2.7 LEADING INDICATOR APPROACH

Product life cycles are becoming shorter and shorter, so it is difficult to obtain enough historical data to forecast demands accurately. One idea that has proven to work well in such situations is the use of *leading indicators*—products that can be used to predict the demands of other, later products because the two products share a similar demand pattern. This approach was introduced by Aytac and Wu (2013) and by Wu et al. (2006), who describe an application of the method at the semiconductor company Agere Systems.

The approach is applied in situations in which a company introduces many related products, such as multiple varieties of semiconductors, cellular phones, or grocery items. The idea is first to group the products into clusters so that all of the products within a cluster share similar attributes. There are several ways to perform this clustering. If one can identify a few demand patterns that all products follow, then it is natural simply to group products sharing the same pattern into the same cluster. For instance, after examining demand data for about 3500 products, Meixell and Wu (2001) find that the products follow six basic demand patterns (i.e., diffusion curves from the Bass model in Section 2.6) and can be grouped into these patterns using statistical cluster analysis. Wu et al. (2006), on the other hand, focus on exogenously defined product characteristics, such as resources, technology group, or sales region, and group the products that have similar characteristics into the same cluster.

The goal is then to identify some potential leading-indicator products within each cluster. A product is a leading indicator if the demand pattern of this product will likely be approximately repeated later by other products in the same cluster. For example, Figure 2.8 depicts the demand for a leading indicator product (solid line) and the total demand for all of the products in the cluster (dashed line). If the leading indicator curve is shifted to the

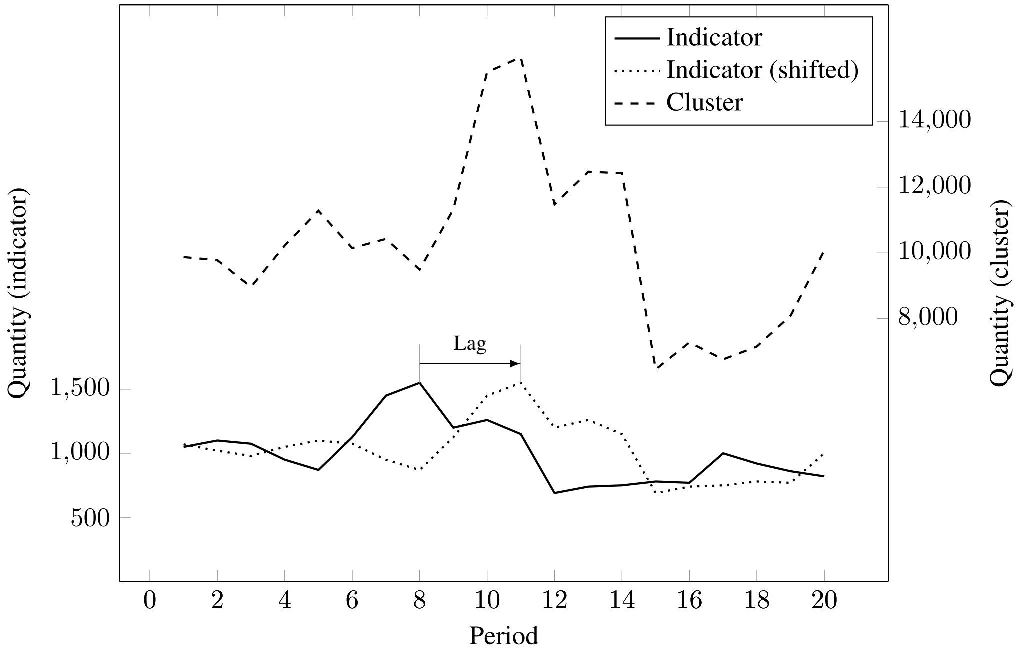


Figure 2.8 An example of a leading-indicator product.

right by three periods (the "lag"), the two curves share a similar structure. Therefore, the leading indicator product provides some basis for predicting the demand of the rest of the products in the cluster. Even though all of the products are on the market simultaneously, the lag provides enough time so that supply chain planning for the products in the cluster can take place based on the forecasts provided by the leading indicator. Of course, correctly identifying the leading indicator is critical.

Wu et al. (2006) suggest the following procedure to identify a leading indicator within a given cluster. Let C be the set of products, i.e., the cluster. Each product $i \in C$ will be treated as a potential leading indicator. Suppose we have historical demand data through period T. Let D_{it} be the observed demand for product i in period t, and let D_t be the total demand for the entire cluster in period t, $t = 1, \ldots, T$. Then leading indicators can be identified using Algorithm 2.1. In line 4 of the algorithm, the correlation ρ_{ik} measures how well the demand of item i over the time interval $[1, T-k]$ predicts the demand of the cluster over $[k+1, T]$.

Once a leading indicator i with time lag k is identified as having a satisfactory correlation coefficient ρ_{ik}, we can forecast the demand for the rest of the product cluster using the demand history from the leading indicator as follows:

1. Regress the demand time-series of product cluster C (excluding i) over $[k+1, T]$ against the time series of the leading indicator over $[1, T-k]$ using the model

$$D_t^{-i} = \beta_0 + \beta_1 D_{i,t-k} \tag{2.54}$$

and determine the optimal regression parameters β_0 and β_1.

Algorithm 2.1 Leading-indicator identification

1: choose $k_{\min}$, $k_{\max}$, $\rho_{\min}$ ▷ Initialization
2: **for all** $i \in C, k \in \{k_{\min}, \ldots, k_{\max}\}$ **do** ▷ Correlation calculation
3: shift product-i demands by k periods
4: calculate ρ_{ik} (correlation between "i lag k" and $C \setminus \{i\}$) as

$$\rho_{ik} \leftarrow \frac{\sum_{t=k+1}^{T}(D_{i,t-k} - \bar{D}_i)(D_t^{-i} - \bar{D}^{-i})}{\sqrt{\sum_{t=k+1}^{T}(D_{i,t-k} - \bar{D}_i)^2 \sum_{t=k+1}^{T}(D_t^{-i} - \bar{D}^{-i})^2}},$$

where D_{it} is the observed demand for product i in period t, $\bar{D}_i$ is its mean over the time interval $[k+1, T]$, D_t^{-i} is the total demand for all products in the cluster excluding i in period t, and $\bar{D}^{-i}$ is its mean over the time interval $[k+1, T]$
5: **end for**
6: **for all** $i \in C, k \in \{k_{\min}, \ldots, k_{\max}\}$ **do** ▷ Identification of leading indicators
7: **if** $\rho_{ik} \geq \rho_{\min}$ **then**
8: label i as leading indicator with lag k
9: **end if**
10: **end for**
11: **if** any leading indicators were found **then**
12: **return** leading indicators and corresponding clusters
13: **else**
14: **for all** C **do** ▷ Reclustering
15: using statistical cluster analysis, subdivide C into clusters based on statistical demand patterns; attributes can include demand mean or SD, shipment frequency, etc.
16: **end for**
17: **go to** 2
18: **end if**

2. For a given month $t > T$ (that is, a month for which we do not have historical data but whose demand we wish to forecast), generate the forecast for the cluster, $\tilde{D}_t^{-i}$, using the time series of the leading indicator i from k periods earlier:

$$\tilde{D}_t^{-i} = \beta_0 + \beta_1 D_{i,t-k}. \tag{2.55}$$

2.8 DISCRETE CHOICE MODELS

2.8.1 Introduction to Discrete Choice

In economics, *discrete choice models* involve choices between two or more discrete alternatives. For example, a customer chooses which of several competing products to buy; a firm decides which technology to use in production; or a passenger chooses which transportation mode to travel by. The set of choices is assumed to be discrete, and the corresponding models are therefore called discrete choice models. (A related set of models, called continuous choice models, assume that the range of choices is continuous. Although these models are not the focus of our discussion, many of the concepts that we describe below are easily transferable to continuous choice models. In fact, discrete choices generally reveal less information about the choice process than continuous ones, so the econometrics of discrete choice is usually more challenging.)

The idea behind discrete choice models is to build a statistical model that predicts the choice made by an individual based on the individual's own attributes as well as the attributes of the available choices. For example, a student's choice of which college to attend is determined by factors relating to the student, including his or her career goals, scholarly interests, and financial situation, as well as factors relating to the colleges, including their reputations and locations. Choice models attempt to quantify this relationship statistically. Rather than modeling the attributes (career goals, scholarly interests, etc.) as independent variables and then predicting the choice as the dependent variable, choice models are at the aggregate (population) level and assume that each decision-maker's preferences are captured implicitly by that model.

At first, it may seem that discrete choice models mainly deal with "which"-type rather than "how many"-type decisions, unlike the other forecasting and demand-modeling techniques described in this chapter. However, discrete choice models can be and have been used to forecast quantities, such as the number and duration of phone calls that households make (Train et al. 1987); the demand for electric cars (Beggs et al. 1981) and mobile telephones (Ida and Kuroda 2009); the demand for planned transportation systems, such as highways, rapid transit systems, and airline routes (Train 1978, Ramming 2001, Garrow 2010)); and the number of vehicles a household chooses to own (McFadden 1984). Choice models estimate the probability that a person selects a particular alternative. Thus, aggregating the "which" decision across the population will give answers to the "how many" questions and can be very useful for forecasting demand.

Discrete choice models take many forms, including binary and multinomial logit, binary and multinomial probit, and conditional logit. However, there are several features that are common to all of these models. These include the way they characterize the choice set, consumer utility, and the choice probabilities. We briefly describe each of these features next. (See Train (2009) for more details about these features.)

The Choice Set: The *choice set* is the set of options that are available to the decision-maker. The alternatives might represent competing products or services, or any other options or items among which the decision-maker must choose. For a discrete choice model, the set of alternatives in the choice set must be *mutually exclusive*, *exhaustive*, and *finite*. The first two requirements mean that the set must include all possible alternatives (so that the decision-maker necessarily does make a choice from within the set) and that choosing one alternative means not choosing any others (so one alternative from the set dominates all other options for the decision-maker). The third requirement distinguishes discrete choice analysis from, say, linear regression analysis in which the dependent variable can (theoretically) take an infinite number of values.

Consumer Utility: Suppose there are N decision-makers, each of whom must select an alternative from the choice set I. A given decision-maker n would obtain a certain level of *utility* from alternative $i \in I$; this utility is denoted U_{ni}. Discrete choice models usually assume that the decision-maker is a utility maximizer. That is, he will choose alternative i if and only if $U_{ni} > U_{nj}$ for all $j \in I, j \neq i$.

If we know the utility values U_{ni} for all $n \in N$ and all $i \in I$, then it will be very easy for us to calculate which alternative decision-maker n will choose (and therefore to predict the demand for each alternative). However, since in most cases we do not know the utility values perfectly, we must estimate them. Let V_{ni} be our estimate of alternative i's utility for decision-maker n. (The V_{ni} values are called *representative utilities*. We omit a discussion about how these might be calculated; see, for example, Train (2009).) Normally, $V_{ni} \neq U_{ni}$, and we use ϵ_{ni} to denote the random estimation error; that is,

$$U_{ni} = V_{ni} + \epsilon_{ni}. \tag{2.56}$$

Choice Probabilities: Once we have determined the V_{ni} values, we can calculate P_{ni}, the probability that decision-maker n chooses alternative i, as follows:

$$\begin{aligned} P_{ni} &= \mathbb{P}(U_{ni} > U_{nj} \quad \forall j \neq i) \\ &= \mathbb{P}(V_{ni} + \epsilon_{ni} > V_{nj} + \epsilon_{nj} \quad \forall j \neq i) \end{aligned} \tag{2.57}$$

The V_{ni} values are constants. To estimate the probability, then, we need to know the probability distributions of the random variables ϵ_{ni}.

Different choice models arise from different distributions of ϵ_{ni} and different methods for calculating V_{ni}. For instance, the logit model assumes that ϵ_{ni} are drawn iid from a member of the family of generalized extreme value distributions, and this gives rise to a closed-form expression for P_{ni}. (Logit is therefore the most widely used discrete choice model.) The probit model, on the other hand, assumes that ϵ_{ni} come from a multivariate normal distribution (and are therefore correlated, not iid), but the resulting P_{ni} values cannot be found in closed form and must instead be estimated using simulation.

2.8.2 The Multinomial Logit Model

Next we derive the multinomial logit model. (Refer to McFadden (1974) or Train (2009) for further details of the derivation.) "Multinomial" means that there are multiple options from which the decision-maker chooses. (In contrast, binomial models assume there are only two options.) The logit model is obtained by assuming each ϵ_{ni} is independently and identically distributed from the standard Gumbel distribution, a type of generalized extreme value distribution (also known as type I extreme value). The pdf and cdf of the standard Gumbel distribution are given by

$$f(x) = e^{-x}e^{-e^{-x}} \tag{2.58}$$

$$F(x) = e^{-e^{-x}}. \tag{2.59}$$

We can rewrite the probability that decision-maker n chooses alternative i (2.57) as

$$P_{ni} = \mathbb{P}(\epsilon_{nj} < V_{ni} + \epsilon_{ni} - V_{nj} \quad \forall j \neq i). \tag{2.60}$$

Since ϵ_{nj} has a Gumbel distribution, by (2.59) the probability in the right-hand side of (2.60) can be written as

$$e^{-e^{-(\epsilon_{ni}+V_{ni}-V_{nj})}}$$

if ϵ_{ni} is given. Since the ϵ are independent, the cumulative distribution over all $j \neq i$ is the product of the individual cumulative distributions:

$$P_{ni}|\epsilon_{ni} = \prod_{j\neq i} e^{-e^{-(\epsilon_{ni}+V_{ni}-V_{nj})}}.$$

Therefore, we can calculate P_{ni} by conditioning on ϵ_{ni} as follows:

$$\begin{aligned} P_{ni} &= \int (P_{ni}|\epsilon_{ni}) f(\epsilon_{ni}) d\epsilon_{ni} \\ &= \int (P_{ni}|\epsilon_{ni}) e^{-\epsilon_{ni}} e^{-e^{-\epsilon_{ni}}} d\epsilon_{ni} \\ &= \int \left(\prod_{j\neq i} e^{-e^{-(\epsilon_{ni}+V_{ni}-V_{nj})}} \right) e^{-\epsilon_{ni}} e^{-e^{-\epsilon_{ni}}} d\epsilon_{ni}. \end{aligned} \tag{2.61}$$

After some further manipulation (see Problem 2.24), we get

$$P_{ni} = \frac{e^{V_{ni}}}{\sum_j e^{V_{nj}}}. \tag{2.62}$$

(The sum in the denominator is over *all* j, including $j = i$.) Note that the probability that individual n chooses alternative i is between 0 and 1 (as is necessary for a well defined probability). As V_{ni}, the estimate of i's utility for n, increases, so does the probability that n chooses i; this probability approaches 1 as V_{ni} approaches ∞. Similarly, as V_{ni} decreases, so does the probability that n chooses i, approaching 0 in the limit.

The expected number of individuals who will choose product i, $N(i)$, is simply given by

$$N(i) = \sum_{n=1}^{N} P_{ni}. \tag{2.63}$$

Of course, we usually don't know P_{ni} for every individual n, so instead we resort to methods to estimate $N(i)$ without relying on too much data. See Koppelman (1975) for a discussion of several useful techniques for this purpose.

Table 2.5 Estimated utilities V_{ni} for uPhone models for Example 2.11.

Model	Tech-Heads	Mainstream	Casual
10B	0.1	0.6	0.4
10W	-0.2	0.7	0.5
10+B	1.3	0.5	-0.1
10+W	1.1	0.4	0.1

Table 2.6 $\exp(V_{ni})$ values for Example 2.11.

Model	Tech-Heads	Mainstream	Casual
10B	1.11	1.82	1.49
10W	0.82	2.01	1.65
10+B	3.67	1.65	0.90
10+W	3.00	1.49	1.11

Table 2.7 Choice probabilities P_{ni} and segment sizes for Example 2.11.

Model	Tech-Heads	Mainstream	Casual
10B	0.13	0.26	0.29
10W	0.10	0.29	0.32
10+B	0.43	0.24	0.18
10+W	0.35	0.21	0.21
Segment size	0.3 M	1.7 M	0.4 M

☐ **EXAMPLE 2.11**

Pear Computer is about to launch model 10 of its popular smart phone, the uPhone. The company is planning four new versions of the uPhone: the uPhone 10 white and black (abbreviated as models 10W and 10B, respectively) and the uPhone 10+ white and black (models 10+W and 10+B). The company has segmented the market into three categories, which they call Tech Heads, Mainstream Users, and Casual Users. Based on market research, Pear Computer has estimated the utilities V_{ni} of each category for each phone model as given in Table 2.5.

The company wishes to know the probability that a user of each market segment will choose each model. We will assume the estimation errors have a Gumbel distribution.

Table 2.6 lists the values of $\exp(V_{ni})$ for all n and i. From these, we can estimate the probabilities P_{ni} as shown in Table 2.7. Note that for a given market segment, the probabilities for the four models sum to 1 (except for rounding error) since we are assuming each consumer will choose exactly one of the models. If we wanted to model the situation in which a consumer may choose not to purchase any uPhone, then we could add a fifth option representing no purchase.

Table 2.7 also lists the total size of each market segment. From the information in the table, we can estimate the total number of each model sold. For example, Pear

Computer can expect to sell

$$0.13 \times 0.3 + 0.26 \times 1.7 + 0.29 \times 0.4 = 0.60$$

million units of the model 10B. Similarly, the demand forecast is 0.65 M for 10W, 0.60 M for 10+B, and 0.55 M for 10+W. $\square$

We refer the readers to other texts (Ben-Akiva and Lerman 1985, Train 2009) for details about this and other choice models. We next give an example of how discrete choice modeling techniques can be used to estimate demand in a supply chain management setting.

2.8.3 Example Application to Supply Chain Management

Suppose there is a retailer who sells a set I of products. The retailer wishes to estimate the probability that a given customer would be interested in purchasing product i, for $i \in I$, so that he can decide which products to offer. Suppose that the customer follows a multinomial logit choice model, as in Section 2.8.2. The retailer's estimate V_i of the customer's utility U_i for product $i \in I$ is given by

$$U_i = V_i + \epsilon_i. \tag{2.64}$$

(Equation (2.64) is identical to (2.56) except that we have dropped the index n since we are considering only a single customer.) If $i = 0$, then U_i and V_i denote the actual and estimated utility of making no purchase.

For any subset $S \subseteq I$, let $P_i(S)$ denote the probability that the customer will purchase product i, assuming that her only choices are in the set S, and let $P_i(S) = 0$ if $i \notin S$. Let $P_0(S)$ denote the probability that the customer will not purchase any product. Then, from (2.62), we have

$$P_i(S) = \begin{cases} \frac{e^{V_i}}{e^{V_0} + \sum_j e^{V_j}}, & \text{if } i \in S \cup \{0\}, \\ 0, & \text{otherwise.} \end{cases} \tag{2.65}$$

The retailer's objective is to choose which products to offer in order to maximize his expected profit. Suppose that the retailer earns a profit of π_i for each unit of product i sold. Suppose also that the retailer cannot offer more than C products. (C might represent shelf space.) Then the retailer needs to solve the following *assortment problem*:

$$\begin{aligned} \text{maximize} \quad & \sum_{i \in S} \pi_i P_i(S) && (2.66) \\ \text{subject to} \quad & |S| \le C && (2.67) \\ & S \subseteq I && (2.68) \end{aligned}$$

(If there are multiple customers, we can just multiply the objective function by the number of customers, assuming they have identical utilities. For a discussion of handling non-homogenous customers, see Koppelman (1975).) This is a combinatorial optimization problem; the goal is to choose the subset S. This problem is not trivial to solve (though it can be solved efficiently). However, the bigger problem is that the utilities U_i, and hence the probabilities $P_i(S)$, are unknown to the retailer. One option is for the retailer to offer different assortments of products over time, estimate the utilities based on the observed demands for each assortment, and refine his assortment as his estimates improve.

Rusmevichientong et al. (2010) propose such an approach. They introduce a policy that the retailer can follow to generate a sequence of assortments in order to maximize the expected profit over time. The assortment offered in a given period depends on the demands observed in the previous periods. Rusmevichientong et al. (2010) also propose a polynomial-time algorithm to solve the assortment problem itself.

CASE STUDY 2.1 Semiconductor Demand Forecasting at Intel

Wu et al. (2010) describe a collaboration between Intel Corporation and Lehigh University researchers to apply the leading-indicator approach (Section 2.7) to forecast demands for new products in the semiconductor industry.[2] At the time of the collaboration, Intel was the largest semiconductor manufacturer in the world and produced chips for several vertical markets, such as mobile, desktop, and server devices. Forecasting demands for semiconductors is difficult due to their short life cycles, long lead times, and high demand volatility. (For another application of the leading-indicator approach in the semiconductor industry, see Wu et al. (2006).)

The approach developed by the researchers involved two key ideas. The first is that by combining forecasts from multiple diffusion models (including, possibly, the Bass model of Section 2.6), we may get better forecasts than if we simply choose a single diffusion model. The second is that leading indicators can be used to update the forecast obtained from the diffusion models using a Bayesian approach.

In particular, Wu et al. (2010) propose fitting, say, 10 different diffusion models to historical data. The Bass model is one good choice, but there are other similar models such as the Weibull, Skiadas, and simple logistic diffusion models. In particular, if we have already observed T periods of demand data for the new product, we can best-fit the parameters of each diffusion model (see Section 2.6.3) to the historical data and evaluate the accuracy of each model. The poorly performing models can be eliminated (for the Intel study, the list was narrowed down to five), and the remaining models can each be used to produce a forecast for the demands in period $T + 1$ through $T + \tau$, for some desired τ. An error term can be added to the forecast to produce a probability distribution rather than just a point forecast. This distribution is called the *prior distribution*.

Next, leading indicators are identified from older generation products or other available time series. For each leading indicator, we generate a forecast for periods $T + 1, \ldots, T + k$, where k is the lag for that leading indicator. (See Section 2.7.) Then, we fit each diffusion model to this extended time series in which periods $1, \ldots, T$ come from observed data and $T+1, \ldots, T+k$ come from the leading indicator forecast. We then use the diffusion model, with the parameters determined in the previous step, to produce a forecast distribution for periods $T + k + 1, \ldots, T + \tau$. This distribution is called the *sampling distribution*.

Finally, we perform a Bayesian update using the prior and sampling distributions to produce a *posterior distribution* for each diffusion model. These distributions are then combined by taking, for each future time period, a weighted sum of the forecasts

[2] In this and subsequent case studies, we have adapted the original notation to be consistent with the rest of the book. In some cases we have also simplified or made other minor modifications to the models, while striving to maintain the main ideas of the original models.

generated by the various diffusion models. Wu et al. (2010) show both analytically and empirically that this results in a smaller variance of forecast error than any of the individual forecasts.

The team implemented the method for 60 Intel products from the mobile, desktop, and server markets. Wu et al. (2010) report that over 10 monthly forecasting cycles, the new method reduced the 12-month forecast error, as measured by MAPE (see Section 2.3), by 9.7%. Moreover, the accuracy of the 4-month forecast, which is the most important given the products' production cycles, improved by 33%. Intel estimated that this would translate to at least \$1.3 million in increased revenue per product over 4 months due to the improved forecasts leading to fewer stockouts. In addition, the decision-support system built by the team to implement this approach executes quickly, reducing the time required to generate forecasts from approximately 3 days under Intel's old aproach to 2 hours using the new system. The work described by Wu et al. (2010) was a finalist for INFORMS's prestigious Wagner Prize for Excellence in Operations Research Practice; see Butler and Camm (2010).

PROBLEMS

2.1 (Forecasting without Trend) A hospital receives regular shipments of liquefied oxygen, which it converts to oxygen gas that is used for life support. The company that sells the oxygen to the hospital wishes to forecast the amount of liquefied oxygen the hospital will use tomorrow. The number of liters of liquefied oxygen used by the hospital in each of the past 30 days is reported in the file `oxygen.xlsx`.

- **a)** Using a moving average with $N = 7$, forecast tomorrow's demand.
- **b)** Using single exponential smoothing with $\alpha = 0.1$, forecast tomorrow's demand.

2.2 (Forecasting with Trend) The demand for a new brand of dog food has been steadily rising at the local PetMart pet store. The previous 26 weeks' worth of demand (number of bags) are given in the file `dog-food.xlsx`.

- **a)** Using double exponential smoothing with $\alpha = 0.2$ and $\beta = 0.1$, forecast next week's demand. Initialize your forecast by setting $I_t = D_t$ for $t = 1, 2$ and $S_2 = I_2 - I_1$.
- **b)** Using linear regression, forecast next week's demand.

2.3 (Forecasting Cupcake Sales) Karl's Cupcakes recently launched a new variety of cupcake. The weekly demands, measured in dozens, during the first two weeks of sales were $D_1 = 47.2$ and $D_2 = 52.3$.

- **a)** Use double exponential smoothing with $\alpha = 0.1$ and $\beta = 0.2$ to calculate y_3, the forecast made in week 2 for the demand in week 3.
- **b)** Suppose the actual demand in week 3 is 59.4. What is y_4, the forecast made in week 3 for the demand in week 4?

2.4 (Forecasting with Seasonality) A hardware store sells potting soil, the demand for which is highly seasonal and has also exhibited a slight upward trend. The number of bags of soil sold each month for the past 40 months is reported in the file `potting-soil.xlsx`. Using triple exponential smoothing with $\alpha = 0.2$, $\beta = 0.1$, and $\gamma = 0.3$, forecast the demand for May. Initialize your forecast by setting

$$
\begin{aligned}
I_t &= D_t \\
S_t &= I_t - I_{t-1} \\
c_t &= \frac{12D_t}{\sum_{i=1}^{12} D_i}
\end{aligned}
$$

for periods $t = 1, \ldots, 12$. (There are better ways to initialize this method, but this method is simpler.)

2.5 (Forecasting Melon Slicers) Matt's Melon Slicers sells specialized knives for watermelons, the demand for which is highly seasonal, with the majority of the demand occurring during the summer. The company has been selling melon slicers for three years and has calculated the following estimates of the seasonal factors, with each period representing one quarter:

Quarter	t	c_t
Winter	9	0.4
Spring	10	0.8
Summer	11	1.9
Fall	12	0.9

At the end of period 12, the company calculated the following estimates of the base signal and slope: $I_{12} = 642$, $S_{12} = 84$.

a) Calculate y_{13}, the forecast made in period 12 for the demand in period 13.

b) Suppose the demand in period 13 turns out to be 341. Calculate I_{13}, S_{13}, and c_{13}.

2.6 (Forecasting Using Regression) The demand for bottled water at football (aka soccer) matches is correlated to the outside temperature at the start of the match. The file `bottled-water.xlsx` reports the temperature (°C) and number of bottles of water sold for each home match played at a certain stadium for the past two seasons (19 home matches per season).

a) Using these data, build a linear regression model to relate the demand for bottled water to the match-time temperature. What are $\hat{\beta}_0$ and $\hat{\beta}_1$?

b) The temperatures for the next three matches are predicted to be $21.6°$, $27.3°$, and $26.6°$, respectively. Forecast the demand for bottled water at each of these matches.

2.7 (Multiple-Period-Ahead Forecasts) In this chapter, we discussed time-series methods for forecasting the demand one period ahead, i.e., in period $t-1$, we generate a forecast y_t for the demand in period t. Suppose instead that we wish to forecast multiple periods ahead, i.e., in period $t-1$, we generate a forecast $y_{t-1,t+k}$ for the demand in period $t+k$, for $k \geq 0$. Explain how to adapt each of the following methods to handle this case:

a) Moving average

b) Double exponential smoothing

c) Linear regression

2.8 (Forecasting using Machine Learning Methods) Using the data set provided in Problem 2.6, choose a learning-based forecasting method—a tree-based model, SVR, or neural networks—for forecasting bottled water given temperatures. Use your selected method to forecast the demand during matches when the temperatures are 21.6°, 27.3°, and 26.6°. Compare your results with those you obtained using linear regression in Problem 2.6(b).

2.9 (Ridge Regression) Ridge regression introduces an ℓ_2-norm penalty to the objective function of linear regression. Consider a simple version in which we have only a single input ($p = 1$); then we are minimizing

$$\sum_{i=1}^{n} \left(y^i - (\beta_0 + \beta_1 x^i)\right)^2 + \lambda(\beta_0^2 + \beta_1^2),$$

where $\lambda > 0$ is the penalty parameter. Derive closed-form expressions for β_0 and β_1. You may use a matrix representation if you wish.

2.10 (Forecasting Fires) The file `nyc-fires.csv` contains the number of fires responded to by the New York City Fire Department on each day from January 1, 2013 through June 30, 2016 (NYC OpenData 2017). It also contains the high temperature (in $^\circ$F) and the total precipitation (in inches) on the same days (National Oceanic and Atmospheric Administration (NOAA) 2017).

Load the data into MATLAB, Excel, or another software package of your choice. Add a variable called `IsWeekend` that indicates whether each day is a weekend day (Saturday or Sunday). Split the data into two parts, one for 2013–2015 (this will be your training data) and one for 2016 (this will be your testing data).

In this problem, you will build models to predict the number of fires on a given day using the three features (high temperature, precipitation, and weekend (Y/N)). Use only the training data when building your models.

a) Build a linear regression model. Report the coefficients $\hat{\beta}_i$.

b) Build a regression tree model with at most 10 branching nodes. (A branching node is a node that has child nodes.) Include a diagram of your tree.

c) Build an SVR model. Report the coefficients $\boldsymbol{\beta}$ and β_0.

d) For each method in parts (a)–(c), predict the number of fires on each day in the testing data. Report the predicted and actual values and the forecast error for the first 10 records in the testing data. Also report the MSE for each method for the entire testing set.

2.11 (Exponential Smoothing for Retail Sales) The file `retail-sales-data.csv` contains weekly sales data for 99 departments within 45 retail stores over approximately 3 years. This is actual data from a real company but has been anonymized (see Kaggle.com (2017)).

a) Extract the sales data for store 2, department 93. Determine the most appropriate form of exponential smoothing (single, double, or triple) and apply that method to forecast the sales. Use 0.15 for all of the smoothing constants (α, β, and/or γ). Begin forecasting at the earliest period you can. (For example, in double exponential smoothing the forecasts begin in period 3.) Report the MSE, MAD, and MAPE for your forecasts. Plot the actual and forecast sales on a single plot.

b) Repeat part (a) for store 3, department 60.

c) Repeat part (a) for store 1, department 16.

2.12 (Mean and Variance of Exponential Smoothing Forecast Error) Prove equations (2.31) and (2.32).

2.13 (Forecasting Simulation) Consider a product whose daily demand follows (2.30) with $\mu = 40$ and $\sigma = 6$.

a) Build a spreadsheet simulation of the demand process, as well as a moving average forecast of order 5. Simulate the system for at least 500 periods. Report the MSE and MAD of the forecast. Also calculate the standard deviation of the forecast error. How accurate is the approximation given in (2.28) for your simulated values?

b) Repeat part (a) for an exponential smoothing forecast with constant $\alpha = 0.1$.

c) Based on the results of parts (a) and (b), does one forecasting method appear to work better than the other?

2.14 (Bass Diffusion for LPhone) HCT, an Asian manufacturer of a new 4G cell phone, the LPhone 5, is planing to enter the U.S. market, and they are in the process of signing a contract with a third-party logistics (3PL) provider in which they must specify the size of the warehouse they want to rent from the 3PL. HCT wants to forecast the total sales of the LPhone 5, as well as the time at which the LPhone 5 reaches its peak sales. After some thorough market research, HCT has estimated that $p = 0.008$, $q = 0.421$, and $m = 5.8$ million. Calculate when the peak sales will occur and how many LPhone 5 the company will have sold by that point.

2.15 (Bass Diffusion for iPeel) Banana Computer Co. plans to launch its latest consumer electronic device, the iPeel, early next year. Based on market research, it estimates that the market potential for the iPeel is 170,000 units, with coefficients of innovation and imitation of 0.07 and 0.31, respectively.

a) If the iPeel is introduced on January 1, on what date will the sales peak? What will be the demand rate on that date, and how many units will have been sold?

b) On what date will 90% of the sales have occurred?

c) Plot the demand rate and cumulative demand as a function of time.

2.16 (Bass Diffusion for Books) A new novel was published recently, and the demand for it is expected to follow a Bass diffusion process. The publisher decided to print only a limited number of copies, observe the demand for the book for 20 weeks, estimate the Bass parameters, and then undertake a second printing for the remainder of the life cycle of the book using these parameters. The demand for the book during these 20 weeks is reported in the file `novel.xlsx`. Using these data, estimate m, p, and q using the method described in Section 2.6.3.

2.17 (Proof of Corollary 2.2) Prove Corollary 2.2.

2.18 (Influentials and Imitators) Suppose that potential adopters of a given product fall into two distinct segments: *influentials* and *imitators*. Each segment has its own within-segment innovation and imitation parameters and experiences its own Bass-type contagion process. In addition, the influentials can exert a cross-segment influence on the imitators, but not vice-versa. Let θ denote the proportion of influentials in the population of eventual adopters ($0 \le \theta \le 1$), and $\bar{\theta} = 1 - \theta$ denote the proportion of imitators. Let p_i and q_i

denote the within-segment innovation and imitation parameters, respectively, for $i = 1, 2$, where $i = 1$ represents influentials and $i = 2$ represents imitators. Let q_c denote the cross-segment imitation parameter.

a) Write a formula expressing each segment's instantaneous adoption behavior, analogous to (2.42).

b) What is special about the case in which $\theta = 0$ or $\theta = 1$?

c) If there are no pre-release purchases (i.e., $D_1(0) = D_2(0) = 0$), write a formula expressing the cumulative adoption at time t, analogous to (2.43).

2.19 (Demand Diffusion across Multiple Markets) A company plans to introduce a variety of new products to multiple vertical markets. The demands from these verticals are likely to follow different diffusion patterns. The company is interested in combining diffusion models derived from different vertical markets to help characterize the overall market demand. However, they are not sure about whether doing so would introduce additional variances and biases into the forecast. Show that combining forecasts of different diffusion models using weights that are inversely proportional to their forecast variances yields a combined forecast variance that is smaller than the forecast variance of each individual diffusion model.

2.20 (Leading Indicators) A battery manufacturer produces a large number of models of lithium-ion batteries for use in computers and other electronic devices. The products are introduced at different times and follow different demand processes. The company wishes to determine whether some of the products can serve as leading indicators for the rest of the products. The file `batteries.xlsx` contains historical demand data for 25 products for the past 26 weeks.

a) Using Algorithm 2.1 with parameters $k_{\min} = 3$, $k_{\max} = 9$, and $\rho_{\min} = 0.85$, determine all pairs (i, k) such that product i is a leading indicator with lag k. (*Note*: You should not need to recluster the products.)

b) Using one of the (i, k) you found in part (a), forecast the demand for the rest of the cluster in periods 27 and 28.

2.21 (Discrete Choice with Uniform Errors) Suppose that, in the discrete choice model, the estimation error ϵ_{ni} has a $U[-1, 1]$ distribution for all n and i. Write an expression for P_{ni}, analogous to (2.61). Your expression may include ϵ_{ni}, V_{ni}, and V_{nj}, but not ϵ_{nj}.

2.22 (Discrete Choices for Day Care) A university is in the process of choosing a location for a new day care center for its faculty's children. The two options for the location are city A, where the university is located, or city B, a neighboring city known for larger houses but a longer commute. The university wants to estimate the number of faculty with kids who are living or will live in city A during the next 10 years. To that end, the university wishes to estimate the choice probability between the two cities for a typical family. Suppose that the utility a family obtains from living in each city depends only on the average house purchase price, the distance between the city and the campus, and the family's opinion of the convenience and quality of life of each city. The first two of these factors can be observed by the researcher, but the researcher cannot observe the third. The researcher believes that the observed part of the utility is a linear function of the observed factors; in particular, the utility of living in each city can be written as

$$U_A = -\,0.45PP_A - 0.23D_A + \epsilon_A$$

$$U_B = -0.45PP_B - 0.23D_B + \epsilon_B,$$

where the subscripts A and B denote city A and city B, and PP and D are the purchase price and distance. The unobserved component of the utility for each alternative, ϵ_A and ϵ_B, vary across households depending on how each household views the quality and convenience of living in each city. If these unobserved components are distributed iid with a standard Gumbel distribution, calculate the probability that a household will choose to live in city A.

2.23 (Using Discrete Choice to Forecast Movie Sales) Three new movies will be shown at a movie theater this weekend. The theater wishes to estimate the expected number of people who will come to see each movie so they can decide how many screenings to offer, how large a theater each movie should be shown in, and so on. The movie studios that produced the three movies held "sneak peak" screenings of the films and conducted post-movie interviews of the attendees. Based on these interviews, they estimated the utility of each movie based on a viewer's age range. They also estimated the utility of not seeing any movie. These estimated utilities are denoted V_{ni}, although here n refers not to an individual but to a *type* of individual (based on age range). The following table lists the V_{ni} values, as well as the number of people who are considering seeing a movie at that theater this weekend.

	Age Range		
Movie	16–25	26–35	36+
Prognosis Negative	0.22	0.54	0.62
Rochelle, Rochelle	0.49	0.57	0.51
Sack Lunch	0.53	0.31	0.38
No movie	0.10	0.27	0.41
Population	700	1900	1150

a) Assume that the actual utilities U_{ni} differ from the estimated utilities V_{ni} by an additive iid error term that has a standard Gumbel distribution. Using the multinomial logit model of Section 2.8.2, calculate the expected demand for each movie.

b) Now suppose the movie theater doesn't know about the multinomial logit model and assumes that P_{ni} is simply calculated using a weighted sum of the V_{ni} values; that is,

$$P_{ni} = \frac{V_{ni}}{\sum_j V_{nj}}.$$

What are the expected demands for each movie using this method?

2.24 (Proof of (2.62)**)** Prove equation (2.62).

CHAPTER 3

DETERMINISTIC INVENTORY MODELS

3.1 INTRODUCTION TO INVENTORY MODELING

3.1.1 Why Hold Inventory?

Think about some of the products you bought the last time you went to the grocery store. How much of each did you buy? Why did you choose these quantities?

Here are some possible reasons:

1. You bought a gallon of milk but only a pint of cream because you drink much more milk than cream in a week.

2. You bought a six-pack of soda, rather than a single bottle, because you don't want to have to go to the store every time you want to drink a bottle of soda.

3. You bought a "family size" box of cereal, rather than a small box, because larger boxes are more cost-effective (cheaper per ounce) than smaller ones.

4. Although you usually eat one bag of potato chips per week, you bought three bags in case your hungry friends show up unexpectedly one night this week.

5. You asked the store to special-order your favorite brand of gourmet mustard (which it doesn't normally stock), even though you already have a half jar at home, because you know it will take a few weeks before the mustard is delivered.

Fundamentals of Supply Chain Theory, Second Edition. Lawrence V. Snyder and Zuo-Jun Max Shen.

Companion website: www.wiley.com/go/Snyder/SupplyChainTheory

6. Although it would be more cost-effective and convenient to buy 12 rolls of paper towels, you only bought 3, because you don't have enough space to store 12 rolls at home.

7. You bought four boxes of pasta, even though you only eat one box per week, because they were on sale for a greatly reduced price.

8. Even though grapes were on sale, you bought one pound instead of two because you knew the second pound would spoil before you had a chance to eat them.

9. You bought a pound of butter (four sticks), even though you probably won't use more than one stick before your next trip to the store, because butter only comes in 1-pound packages.

All of these decisions affected the amount of inventory of groceries that you have in your home. Aside from the cost you paid to purchase these items, you are also paying a cost simply to hold the inventory (as opposed to buying a single item each time you need it and using it immediately). For example, if you used your credit card to make your purchase, then you are paying a little more interest by buying a six-pack of soda today rather than buying individual bottles throughout the week. If you paid cash, then you are tying up your cash in groceries rather than using it for some other purpose, such as going to the movies, or putting your money in an interest-earning savings account. You are also paying for the physical space required to store your groceries (as part of your rent or mortgage), the energy required to keep refrigerated items cold, and the insurance to protect your grocery investment if your house is burglarized or damaged in a fire.

Companies, too, would prefer not to hold any inventory, since inventory is expensive (even more than it is for you). However, most companies hold some inventory, for the same reasons that you hold inventory of your groceries:

1. Different products are purchased at different rates—the *demand rate*—and therefore require different levels of inventory.

2. There is an inconvenience, and often an expense, associated with placing an order with a supplier (analogous to your trip to the grocery store). For example, there may be an administrative cost to process the order and transmit it to the supplier, or there may be a cost to rent a truck to deliver the products. These are *fixed costs* since they are (roughly) independent of the size of the order, and they make it impractical to place an order each time a single item is needed.

3. Firms often receive *volume discounts* for placing large orders with their suppliers. Volume discounts and fixed costs are both types of *economies of scale*, which make it more cost-effective to order in bulk; that is, to place fewer, larger orders.

4. Demand for most products is random, and often so are lead times and other supply factors, and this *uncertainty* requires firms to hold inventory to ensure that they can satisfy the demand (at least most of the time).

5. After a firm places an order, the products do not arrive until after a (typically nonzero) *lead time*. Since the firm's own customers usually don't want to wait for this lead time, especially in retail settings, the firm must place a replenishment order even when it is still holding some inventory.

6. Warehouses have only a finite amount of *storage capacity*, and this may constrain the size of the firm's order. A related type of capacity (which is less relevant for the grocery example) is *production capacity*: If demand is highly seasonal (e.g., for snowblowers) but production capacity is limited, then the firm may need to produce more in off-peak times (summer) in order to meet the demand during peak times (winter).

7. Suppliers often offer sales and temporary discounts, just like retail stores do, and prices for many products (especially commodities) vary constantly. In response to both types of *price fluctuations*, firms buy large quantities when prices are low and hold goods in inventory until they're needed.

8. Some inventory is *perishable*, so firms must limit the quantity they buy to avoid being saddled with unusable inventory.

9. Many products are available only in fixed *batch sizes* such as cases or pallets, and the firm is forced to order in increments of those units.

These are all reasons that firms plan to hold inventory. In addition, firms may hold *unplanned* inventory—for example, inventory of products that have become obsolete sooner than expected.

Firms may hold inventory of goods at all stages of production—raw materials, components, work-in-process, and finished goods. The latter types of inventory are usually made by the firm, rather than ordered from a supplier, but similar issues still arise—for example, there may be a fixed cost to initiate a production run, it may be cheaper per unit to produce large batches, the processing time may be uncertain, and so on. In fact, although we tend to discuss inventory models as though the firm is buying a product from an outside supplier, most inventory models apply equally well to production systems, in which case we are deciding how much to produce, rather than how much to order, and the "ordering" costs are really production costs.

3.1.2 Classifying Inventory Models

Mathematical inventory models can be classified along a number of different dimensions:

- *Demand.* Is demand deterministic or stochastic? Does the rate stay the same all the time or does it vary over time—say, from season to season?

- *Lead time.* Is production or delivery instantaneous, or is there a positive lead time? Is the lead time deterministic or stochastic?

- *Review type.* Is inventory assessed *continuously* or *periodically*? In continuous-review models, the inventory is constantly monitored, and an order is placed whenever a certain condition is met (for example, the inventory level falls below a given value). In periodic-review models, the inventory is only checked every time period (say, every week), and an order is placed if the reorder condition is met. In periodic-review models, we usually assume that demands occur at a single instant during the period, even though they may really occur continuously throughout it.

- *Planning horizon. Finite-horizon* models consider a finite number of periods or time units, while *infinite-horizon* models assume the planning horizon extends forever.

Although it is unrealistic to assume that the firm will continue operating the same system, under the same conditions, forever, infinite-horizon models are often more tractable than finite-horizon ones and are therefore quite common.

- *Stockout type.* If demand exceeds supply, how is the excess demand handled? Most models consider either *backorders*, in which case excess demand stays on the books until it can be satisfied from a future shipment, or *lost sales*, in which case excess demands are simply lost—the customer takes her business elsewhere. In retail settings, it is usually more accurate to assume lost sales, whereas backorders are more common in business-to-business settings.

- *Ensuring good service.* Some models ensure that not too many stockouts occur by including a penalty in the cost function for each stockout. Others include a constraint on the allowable percentage of demands that may be stocked out. The former approach often leads to more tractable models, but it can be very difficult to quantify the cost of a stockout; therefore, service-level constraints are common in practice.

- *Fixed cost.* Some inventory models include a fixed cost to place an order, while others do not. The presence and magnitude of a fixed cost determines whether the firm places many small orders or few large orders. Moreover, inventory models with fixed costs are often more difficult to analyze and solve than those without, so we often ignore the fixed cost in modeling an inventory system even if one is present in the real system.

- *Perishability.* Can inventory be held across multiple time periods, or is it perishable? Perishable items include not just foods, but also fresh flowers and medicine (which will spoil), high-tech products (which will become obsolete), and newspapers and airline tickets (which have a deadline after which they can't be sold).

Like all mathematical models, inventory models must balance two competing factors—realism and tractability. In many cases, it is more accurate to assume one thing but easier to assume the opposite. For example, many inventory models are much more mathematically tractable if we assume backorders, so we might do so even if we are modeling inventory at a retail store, for which the lost-sales assumption is more accurate. Similarly, it is often convenient to assume lead times are zero even though they rarely are in practice. If the lead time is short compared to the order cycle—for example, if the firm places monthly orders and the lead time is 2 days—this assumption may not hurt the model's accuracy too much. Modeling is as much an art as a science, and part of modeling process involves determining both the cost (in terms of realism) and the benefit (in terms of tractability) of "assuming away" a given real-life factor.

3.1.3 Costs

The goal of most inventory models is to minimize the cost (or maximize the profit) of the inventory system. Four types of costs are most common:

- *Holding cost.* This represents the cost of actually keeping the inventory on hand. Like the costs associated with storing your groceries, the holding cost includes the

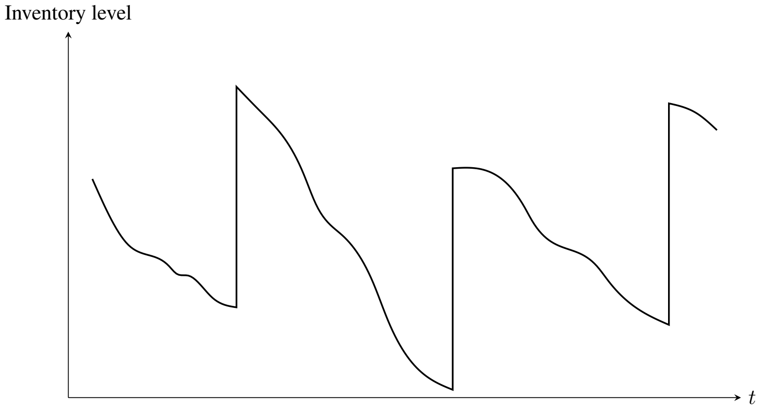


Figure 3.1 Inventory level curve.

cost of storage space, taxes, insurance, breakage, theft, and, most significantly, opportunity cost—the money the firm could be earning if it didn't have its capital tied up in inventory. The holding cost is often expressed as a percentage of the value of the product per year. For example, the holding cost might be 25% per year. If the item costs \$100, then it costs \$1562.50 to hold 250 items for 3 months ($1562.50 = 0.25 \cdot 100 \cdot 250 \cdot (3/12)$). We will usually use h to represent the holding cost per item per unit time.

In reality, the inventory level is not constant but fluctuates over time, as pictured in Figure 3.1. Here, the holding cost is the area under the curve times h, so we would use integration to compute it. In some of the inventory models discussed in this book, the inventory "curve" is made up of straight lines, so computing the area is easy.

- *Fixed cost.* This is the cost to place an order, independent of the size of the order. It is sometimes called the *setup cost*, and we will usually denote it by K. The fixed cost accounts for the administrative cost of placing an order, the cost of using a truck to deliver the product, and so on.

- *Purchase cost.* This is the cost per unit to buy and ship the product, generally denoted by c. (It is also sometimes known as the *variable cost* or *per-unit cost*.) Therefore, the total order cost (fixed + purchase) to order x units is given by

$$\begin{cases} 0, & \text{if } x = 0 \\ K + cx, & \text{if } x > 0. \end{cases}$$

One picky but worthwhile note: If there is a nonzero lead time, then we typically assume that the firm pays the purchase cost c when the order arrives, not when it is placed. This assumption doesn't affect the total purchase cost per year (unless we're modeling the time value of money), but it does affect the holding cost if h is a function of c: If the firm was to pay the purchase cost when the order is placed, its capital would be tied up during the lead time, but this would not be accurately reflected in the holding cost.

- *Stockout cost.* This is the cost of not having sufficient inventory to meet demand, also called the *penalty cost* or *stockout penalty*, and is denoted by p. If excess demand is backordered, the penalty cost includes bookkeeping costs, delay costs, fines for missing promised delivery dates, and—most significantly—*loss of goodwill* (the potential loss of future business since the customer is unhappy). If excess demand is lost, the penalty cost also includes the lost profit from the missed sale. The penalty is generally charged per unit of unmet demand. If excess demand is backordered, the penalty may be proportional to the amount of time the backorder is on the books before it is filled, or (less commonly) it may be a one-time penalty charged when the demand is backordered.

3.1.4 Inventory Level and Inventory Position

There are several measures that we use to assess the amount of inventory in the system at any given time. *On-hand inventory* (OH) refers to the number of units that are actually available at the stocking location. *Backorders* (BO) represent demands that have occurred but have not been satisfied. Generally, it's not possible for the on-hand inventory *and* the backorders to be positive at the same time.

The *inventory level* (IL) is equal to the on-hand inventory minus backorders:

$$IL = OH - BO.$$

If $IL > 0$, we have on-hand inventory, and if $IL < 0$, we have no units on hand but we do have backorders. Therefore, we can write

$$\begin{aligned} OH &= IL^+ \\ BO &= IL^-, \end{aligned}$$

where $x^+ = \max\{x, 0\}$ and $x^- = |\min\{x, 0\}|$. (Be warned: Some authors use $x^- = \min\{x, 0\}$.)

It seems reasonable to think of IL as the relevant measure to consider when making ordering decisions—we look at the shelves, see how much inventory we have, and place an order if there's not enough. But IL by itself does not give us enough information to make good ordering decisions. For instance, suppose the inventory level is 5, you're expecting a demand of 50 next week, and there's a lead time of 4 weeks. How much should you order? The answer depends on how much you've already ordered—i.e., how much is "in the pipeline," ordered but not received. Such items are called *on order* (OO). Therefore, we usually make ordering decisions based on the *inventory position* (IP), which equals the inventory level plus items on order:

$$IP = OH - BO + OO.$$

The distinction between inventory level and inventory position is subtle but important. Typically, we use inventory position to make ordering decisions, but holding and backorder costs are assessed based on inventory level. If the lead time is zero, then $OO = 0$ and $IL = IP$.

3.1.5 Roadmap

In this chapter and the next three, we will explore some classical inventory models and a few of their variants. This chapter discusses deterministic models—first a continuous-review model, the economic order quantity (EOQ) model, perhaps the oldest and best-known mathematical inventory model (Section 3.2), and some of its extensions; and then a periodic-review model, the Wagner–Whitin model (Section 3.7). Then, Chapters 4 and 5 discuss stochastic models. The models in all three of these chapters make inventory decisions for a single stage (location). Multistage models are considered in Chapter 6.

The models discussed in this chapter are sometimes known as *economic lot size problems*. In fact, there is some inconsistency about how this term is used in the literature. Some authors refer to the EOQ model (Section 3.2) as *the* economic lot size model. Other authors refer to the Wagner–Whitin model (Section 3.7) as *the* economic lot size model. More generally, the term can be used to refer to any model in which an optimal lot size must be determined, typically under deterministic demand. To avoid confusion, we will avoid this term and instead use the names of the individual models discussed.

3.2 CONTINUOUS REVIEW: THE ECONOMIC ORDER QUANTITY PROBLEM

3.2.1 Problem Statement

The *economic order quantity* (EOQ) problem is one of the oldest and most fundamental inventory models; it was first introduced by Harris (1913). The goal is to determine the optimal amount to order each time an order is placed to minimize the average cost per year. (We'll express everything per year, but the model could just as easily be per month or any other time period.)

We assume that demand is deterministic and constant with a rate of λ units per year. Stockouts are not allowed—we must always order enough so that demand can be met. Since demand is deterministic, this is a plausible assumption. The lead time is 0—orders are received instantaneously. There is a fixed cost K per order, a purchase cost c per unit ordered, and an inventory holding cost h per unit per year. There is no stockout penalty since stockouts are not allowed.

The inventory level[1] evolves as follows. Assume that the on-hand inventory is 0 at time 0; we place an order at time 0, and it arrives instantaneously. The inventory level then decreases at a constant rate λ until the next order is placed, and the process repeats.

Any optimal solution for the EOQ model has two important properties:

- *Zero-inventory ordering (ZIO) property.* Since the lead time is 0, it never makes sense to place an order when there is a positive amount of inventory on hand—we only place an order when the inventory level is 0.

- *Constant order sizes.* If Q is the optimal order size at time 0, it will also be the optimal order size every other time we place an order since the system looks the same every time the inventory level hits 0. Therefore, the order size is the same every time an order is placed.

(You should convince yourself that these properties are indeed optimal.) The inventory level is pictured as a function of time in Figure 3.2. T is called the *cycle length*—the amount

[1]Since the lead time is 0, the inventory position is equal to the inventory level at all times.

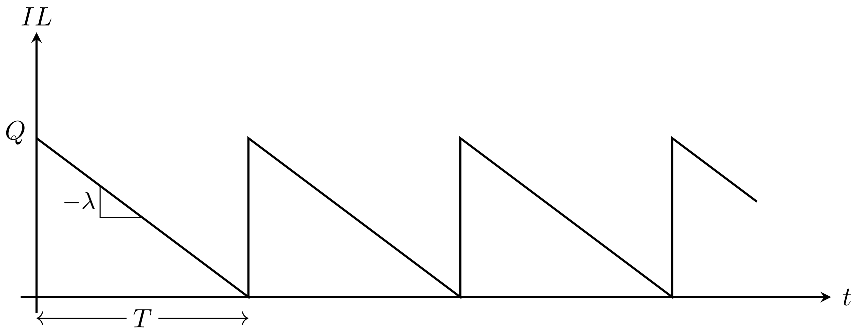


Figure 3.2 EOQ inventory level curve.

of time between orders—and it relates to the order quantity Q and λ by the equation

$$T = \frac{Q}{\lambda}.$$

3.2.2 Cost Function

We want to find the optimal Q to minimize the average annual cost. (We say "average" annual cost since the actual cost in any given year may fluctuate a bit as the sawtooth pattern falls slightly differently across the start of each year.) Note that minimizing the annual cost is not the same as minimizing the cost per cycle; minimizing the cost per cycle would mean choosing very tiny order quantities. The key trade-off is between fixed cost and holding cost: If we use a large Q, we'll place fewer orders and hold more inventory (small fixed cost but large holding cost), whereas if we use a small Q, we'll place more orders and hold less inventory (large fixed cost but small holding cost).

The strategy for solving the EOQ is to express the average annual cost as a function of Q, then minimize it to find the optimal Q.

Order Cost: Each order incurs a fixed cost of K. It also incurs a purchase cost of c per unit ordered, but this cost is irrelevant for the optimization problem at hand—that is, the optimal value of Q does not depend on c. (Why?) Therefore, we'll ignore the per-unit cost c in our analysis. Since the time between orders is T years, the order cost per year is

$$\frac{K}{T} = \frac{K\lambda}{Q}. \tag{3.1}$$

Holding Cost: The average inventory level in a cycle is $Q/2$, so the average amount of inventory per year is $Q/2 \cdot 1$ year $= Q/2$. (Another way to think about this is that the area of a triangle in the inventory curve in Figure 3.2 is $QT/2$, and there are $1/T$ cycles per year, so the total area under the inventory curve for 1 year is $QT/2 \cdot 1/T = Q/2$.) Therefore, the average annual holding cost is

$$\frac{hQ}{2}. \tag{3.2}$$

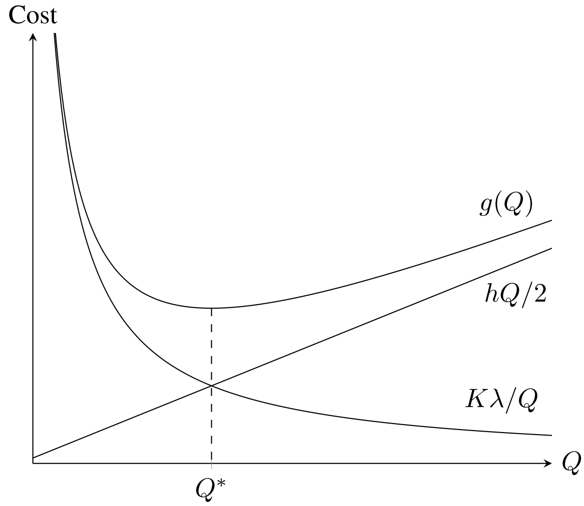


Figure 3.3 Fixed, holding, and total costs as a function of Q.

Total Cost: Combining (3.1) and (3.2), we get the total average annual cost, denoted $g(Q)$:

$$g(Q) = \frac{K\lambda}{Q} + \frac{hQ}{2}. \tag{3.3}$$

The fixed, holding, and total cost curves are plotted as a function of Q in Figure 3.3.

3.2.3 Optimal Solution

The optimal Q can be obtained by taking the derivative of $g(Q)$ and setting it to 0:

$$\begin{aligned}\frac{dg(Q)}{dQ} &= -\frac{K\lambda}{Q^2} + \frac{h}{2} = 0 \\ &\implies Q^2 = \frac{2K\lambda}{h} \\ &\implies Q^* = \sqrt{\frac{2K\lambda}{h}}.\end{aligned} \tag{3.4}$$

Q^* is known as the *economic order quantity*. ("Economic" is just another word for "optimal.") We should also take a second derivative to verify that $g(Q)$ is convex (and thus the first-order condition yields a minimum, not a maximum):

$$\frac{d^2g(Q)}{dQ^2} = \frac{2K\lambda}{Q^3} > 0,$$

as desired.

Note that in Figure 3.3, we drew the optimal order quantity Q^* at the intersection of the fixed and holding cost curves. This was not an accident. Of course, in general, it is not true that the minimum of the sum of two functions occurs where the two functions intersect, but it happens to be true for the EOQ. Why? The curves intersect when

$$\frac{K\lambda}{Q} = \frac{hQ}{2} \implies \frac{K\lambda}{Q^2} = \frac{h}{2}.$$

This is exactly the condition obtained by setting the first derivative to 0. Thus, the fixed and holding costs should always be balanced. If the fixed cost $K\lambda/Q$ is greater than the holding cost $hQ/2$, then Q is not optimal; we should be ordering less frequently and holding more inventory. (And vice versa.)

Another way to see that the fixed and holding costs are equal in the optimal solution is to note that the product of the two terms in (3.3) is

$$\frac{K\lambda}{Q} \cdot \frac{hQ}{2} = \frac{K\lambda h}{2},$$

a constant. In general, when two quantities multiply to a constant, their sum is minimized when the quantities are equal. Another non-calculus-based proof is given in Problem 3.21.

It should also be noted that, although we ignored the per-unit cost c in this analysis, c does influence Q^* indirectly if h is a function of c.

The optimal cost can be expressed as a function of the parameters by plugging the optimal Q^* into $g(Q)$:

$$\begin{aligned} g(Q^*) &= \frac{K\lambda}{\sqrt{\frac{2K\lambda}{h}}} + \frac{h}{2}\sqrt{\frac{2K\lambda}{h}} \\ &= \sqrt{\frac{K\lambda h}{2}} + \sqrt{\frac{K\lambda h}{2}} \\ &= \sqrt{2K\lambda h}. \end{aligned} \tag{3.5}$$

It's nice that the optimal cost has such a convenient form. This is not true for many other problems. The ability to express $g(Q^*)$ in closed form allows us to learn about structural properties of the EOQ and related models, such as the power-of-two policies discussed in Section 3.3, as well as to embed the EOQ into other, richer models, such as the location model with risk pooling (LMRP) in Section 12.2.

The optimal EOQ solution and its cost are summarized in the next theorem, whose proof follows from arguments already made above.

Theorem 3.1 *The optimal order quantity in the EOQ model is given by*

$$Q^* = \sqrt{\frac{2K\lambda}{h}} \tag{3.6}$$

and its cost is given by

$$g(Q^*) = \sqrt{2K\lambda h}. \tag{3.7}$$

Using Theorem 3.1, we can make some statements about how the solution changes as the parameters change:

- As h increases, Q^* decreases, since larger holding cost $\Longrightarrow$ it's more expensive to hold inventory $\Longrightarrow$ order smaller quantities more frequently

- As K increases, Q^* increases, since it's more expensive to place orders $\Longrightarrow$ we place fewer of them, with larger quantities

- As c increases, Q^* decreases if h is proportional to c (and stays the same if they are independent)

- As λ increases, Q^* increases

Obviously, if any of the costs increase, then $g(Q^*)$ will increase. If λ increases, $g(Q^*)$ will increase, as well. This does not mean that the firm prefers small demand, however. Remember that the EOQ only reflects costs, not revenues; the increased cost of large λ would be outweighed by the increased revenue.

☐ **EXAMPLE 3.1**

Joe's Corner Store sells 1300 candy bars per year. It costs \$8 to place an order to the candy bar supplier. Each candy bar costs the store \$0.75. Holding costs are estimated to be 30% per year. What is the optimal order quantity?

We have $h = 0.3 \cdot 0.75 = 0.225$, so

$$Q^* = \sqrt{\frac{2K\lambda}{h}} = \sqrt{\frac{2 \cdot 8 \cdot 1300}{0.225}} = 304.1.$$

The optimal cycle time is

$$T^* = \frac{Q^*}{\lambda} = \frac{304.1}{1300} = 0.23.$$

So the store should order 304.1 candy bars every 0.23 years, or approximately four times per year. The optimal cost is

$$\sqrt{2K\lambda h} = \sqrt{2 \cdot 8 \cdot 1300 \cdot 0.225} = 68.41.$$

If we must order in integer quantities, then we need to round Q^* down and up and check the cost of each:

$$\begin{aligned} g(304) &= \frac{8 \cdot 1300}{304} + \frac{0.225 \cdot 304}{2} = 68.4105 \\ g(305) &= \frac{8 \cdot 1300}{305} + \frac{0.225 \cdot 305}{2} = 68.4108, \end{aligned}$$

so we should order 304. ☐

3.2.4 Sensitivity Analysis

Suppose the firm did not want to order Q^* exactly. For example, it might need to order in multiples of 10 ($Q = 10n$), or it might want to order every month ($T = 1/12$). How much more expensive is a suboptimal solution? It turns out that the answer is "not much," and that we can determine the exact percentage increase in cost using a very simple formula.

Theorem 3.2 *Suppose Q^* is the optimal order quantity in the EOQ model. Then for any $Q > 0$,*

$$\frac{g(Q)}{g(Q^*)} = \frac{1}{2}\left(\frac{Q^*}{Q} + \frac{Q}{Q^*}\right). \tag{3.8}$$

Proof.

$$\frac{g(Q)}{g(Q^*)} = \frac{\frac{K\lambda}{Q} + \frac{hQ}{2}}{\sqrt{2K\lambda h}}$$

$$
\begin{aligned}
&= \frac{K\lambda}{Q\sqrt{2K\lambda h}} + \frac{hQ}{2\sqrt{2K\lambda h}} \\
&= \frac{1}{Q}\sqrt{\frac{K\lambda}{2h}} + \frac{Q}{2}\sqrt{\frac{h}{2K\lambda}} \\
&= \frac{1}{2Q}\underbrace{\sqrt{\frac{2K\lambda}{h}}}_{=Q^*} + \frac{Q}{2}\underbrace{\sqrt{\frac{h}{2K\lambda}}}_{=1/Q^*} \\
&= \frac{1}{2}\left(\frac{Q^*}{Q} + \frac{Q}{Q^*}\right)
\end{aligned}
$$

■

The right-hand side of (3.8) grows slowly as Q deviates more from Q^*, meaning that the EOQ is not very sensitive to errors in Q. For example, if we order twice as much as we should ($Q = 2Q^*$), the error is 1.25 (25% more expensive than optimal). If we order half as much ($Q = Q^*/2$), the error is also 1.25.

Theorem 3.2 ignores the per-unit cost c. If we include the annual cost $c\lambda$ in the numerator and denominator of (3.8), then the percentage increase in cost would be even smaller (and the expressions would not simplify as nicely).

□ **EXAMPLE 3.2**

Suppose Joe's Corner Store (Example 3.1) ordered 250 candy bars per order instead of the optimal 304.1. How much would the cost increase as a result of this suboptimal solution?

$$\frac{g(Q)}{g(Q^*)} = \frac{1}{2}\left(\frac{304.1}{250} + \frac{250}{304.1}\right) = 1.019$$

So this solution would cost 1.9% more than the optimal solution. (You can also confirm this by calculating $g(250)$ explicitly and comparing it to $g(Q^*)$.) □

3.2.5 Order Lead Times

We assumed the lead time is 0. What if the lead time was positive—say, L years? The optimal solution doesn't change—we just place our order L years before it's needed. For example, if $L = 1$ month = 1/12 years, then the order should be placed 1/12 years before the inventory level reaches 0. It's generally more convenient to express this in terms of the *reorder point* (r). When the inventory level reaches r, an order is placed. How do we compute r? Well, r should be equal to the amount of product demanded during the lead time, or

$$r = \lambda L. \tag{3.9}$$

□ **EXAMPLE 3.3**

In Example 3.1, if $L = 1/12$, the store should place an order whenever the inventory level reaches $r = 1300 \cdot (1/12) = 108.3$. □

3.3 POWER-OF-TWO POLICIES

From Section 3.2.3, we know that the optimal solution to the EOQ model is $Q^* = \sqrt{2K\lambda/h}$. We also know that the order interval T is given by $T = Q/\lambda$, so the optimal order interval is $T^* = \sqrt{2K/\lambda h}$. But what if T^* is some inconvenient number? How can we place an order, for example, every $\sqrt{10}$ weeks? In this section, we discuss *power-of-two policies*, in which the order interval is required to be a power-of-two multiple of some *base period*. The base period may be any time period—week, day, work shift, etc. If the base period is a day (say), then the power-of-two restriction says that orders can be placed every 1 day, or every 2 days, or every 4 days, or every 8 days, and so on, or every 1/2 day, or every 1/4 day, and so on. Policies based on a convenient base period such as days or months are more convenient to implement than those involving base periods like $\sqrt{10}$. We already know that the EOQ model is relatively insensitive to deviations from the optimal solution from Theorem 3.2. Our goal is to determine exactly how much more expensive a power-of-two policy is than the optimal policy.

Power-of-two policies have another advantage over the optimal EOQ policy: They make coordination easier at a central warehouse. If retailers each order according to their own EOQ policies, the warehouse will see a chaotic mess of order times. If, instead, each retailer follows a power-of-two policy with the same base period, the warehouse will see orders line up nicely, making its own inventory planning easier. The problem of finding optimal order intervals in this setting is one version of a problem known as the *one warehouse, multiretailer (OWMR) problem*. The optimal policy for the OWMR problem is not known, but it has been shown that power-of-two policies are very close to optimal (Roundy 1985, Muckstadt and Roundy 1993).

3.3.1 Analysis

The problem statement is exactly as in the EOQ model (see Section 3.2.1). In addition, we assume there is some base planning period T_B. The actual reorder interval chosen must be of the form

$$T = T_B 2^k \tag{3.10}$$

for some $k \in \{\ldots, -2, -1, 0, 1, 2, \ldots\}$. We need to determine (1) the best power-of-two policy, i.e., the best value of k, and (2) how far from optimal this policy is.

From the EOQ model, we know that the optimal order interval is

$$T^* = \sqrt{\frac{2K}{\lambda h}}. \tag{3.11}$$

Let $f(T)$ be the EOQ cost if an order interval of T is chosen, ignoring the per-unit cost; that is,

$$f(T) = \frac{K}{T} + \frac{h\lambda T}{2}. \tag{3.12}$$

(This follows from substituting $Q = T\lambda$ in the EOQ cost function (3.3).) One can easily verify that f is convex, so the optimal k in (3.10) is the smallest integer k satisfying

$$f(T_B 2^k) \leq f(T_B 2^{k+1}), \tag{3.13}$$

that is,

$$\begin{aligned}
\frac{K}{T_B 2^k} + \frac{h\lambda}{2} T_B 2^k \le \frac{K}{T_B 2^{k+1}} + \frac{h\lambda}{2} T_B 2^{k+1} \\
\iff \frac{K}{T_B 2^{k+1}} \le \frac{h\lambda}{2} T_B 2^k \\
\iff \frac{K}{h\lambda} \le (T_B 2^k)^2 \\
\iff \frac{1}{\sqrt{2}} T^* = \sqrt{\frac{K}{h\lambda}} \le T_B 2^k.
\end{aligned} \tag{3.14}$$

Therefore, the optimal power-of-two order interval is $\hat{T} = T_B 2^k$, where k is the smallest integer satisfying (3.14).

3.3.2 Error Bound

Theorem 3.3 *If $\hat{T}$ is the optimal power-of-two order interval and T^* is the optimal (not necessarily power-of-two) order interval, then*

$$\frac{f(\hat{T})}{f(T^*)} \le \frac{3}{2\sqrt{2}} \approx 1.06.$$

In other words, the cost of the optimal power-of-two policy is no more than 6% greater than the cost of the optimal (non-power-of-two) policy. This holds for any choice of the base period T_B.

Proof. Since k is the smallest integer satisfying (3.13), we have

$$\begin{aligned}
f(T_B 2^{k-1}) > f(T_B 2^k) \\
\iff \frac{K}{T_B 2^k} > \frac{h\lambda}{2} T_B 2^{k-1} \\
\iff \sqrt{\frac{4K}{h\lambda}} > T_B 2^k,
\end{aligned}$$

or

$$\hat{T} < \sqrt{2} T^*. \tag{3.15}$$

Together, (3.14) and (3.15) imply that the optimal power-of-two order interval $\hat{T}$ must be in the interval $[\frac{1}{\sqrt{2}}T^*, \sqrt{2}T^*)$. Note that this is true for *any* base period T_B. Now, using (3.11) and (3.12),

$$\begin{aligned}
f\left(\frac{1}{\sqrt{2}} T^*\right) &= \frac{\sqrt{2}K}{T^*} + \frac{h\lambda}{2} \frac{1}{\sqrt{2}} T^* \\
&= \frac{\sqrt{2}K}{\sqrt{\frac{2K}{\lambda h}}} + \frac{h\lambda}{2} \frac{1}{\sqrt{2}} \sqrt{\frac{2K}{\lambda h}}
\end{aligned}$$

$$= \frac{3}{2\sqrt{2}}\sqrt{2K\lambda h}$$
$$= \frac{3}{2\sqrt{2}}f(T^*).$$

Similarly,

$$\begin{aligned} f(\sqrt{2}T^*) &= \frac{K}{\sqrt{2}T^*} + \frac{h\lambda}{2}\sqrt{2}T^* \\ &= \frac{1}{\sqrt{2}}\sqrt{\frac{K\lambda h}{2}} + \frac{\sqrt{2}}{2}\sqrt{2K\lambda h} \\ &= \frac{3}{2\sqrt{2}}\sqrt{2K\lambda h} \\ &= \frac{3}{2\sqrt{2}}f(T^*). \end{aligned}$$

Since f is convex and the optimal $\hat{T}$ lies somewhere between $\frac{1}{\sqrt{2}}T^*$ and $\sqrt{2}T^*$,

$$\frac{f(\hat{T})}{f(T^*)} \le \frac{3}{2\sqrt{2}} \approx 1.06.$$

■

Since we don't know precisely where $\hat{T}$ falls in the range $[\frac{1}{\sqrt{2}}T^*, \sqrt{2}T^*)$, this is only a worst-case bound that occurs on the endpoints of the range. If $\hat{T}$ falls somewhere in the middle of the range, the power-of-two policy may be even better than 6% above optimal. In fact, if we assume that $\hat{T}$ is uniformly distributed in the range, we get an expected bound of only 2%:

Theorem 3.4 *Assuming that the optimal power-of-two order interval $\hat{T}$ is uniformly distributed in the range $[\frac{1}{\sqrt{2}}T^*, \sqrt{2}T^*)$,*

$$\frac{\mathbb{E}[f(\hat{T})]}{f(T^*)} \le \frac{1}{\sqrt{2}}\left(\ln 2 + \frac{3}{4}\right) \approx 1.02. \tag{3.16}$$

Proof. Omitted. ■

□ EXAMPLE 3.4

Suppose Joe (owner of Joe's Corner Store, from Example 3.1) must order candy bars in power-of-two multiples of 1 month. What is the optimal power-of-two order interval, and what is the cost ratio versus the optimal (non-power-of-two) solution?

We have $T_B = 1/12$ years. You can confirm that

$$\begin{aligned} f\left(T_B 2^0\right) &= f(0.0833) = 108.19 \\ f\left(T_B 2^1\right) &= f(0.1667) = 72.38 \\ f\left(T_B 2^2\right) &= f(0.3333) = 72.75 \end{aligned}$$

By the convexity arguments above, the optimal power-of-two order interval is $\hat{T} = 0.1667$ years, or every 2 months. The cost ratio is $72.38/68.41 = 1.0580$, within the bound of 1.06. □

3.4 THE EOQ WITH QUANTITY DISCOUNTS

It is common for suppliers to offer discounts based on the quantity ordered. The larger the order, the lower the purchase cost per item. (You may have observed something similar when you shop for groceries. When you buy in bulk, you pay less per unit.) The specific structure for the discounts can take many forms, but two types are most common: *all-units discounts* and *incremental discounts*. Both discount structures use *breakpoints* to determine the purchase price. For example, the supplier may charge \$1 per unit if the firm orders 0–100 units, \$0.90 per unit if the firm orders 100–250 units, and \$0.85 per unit if the firm orders more than 250 units. The two discount structures differ based on how the total purchase cost is determined.

We assume there are n *breakpoints*, denoted $b_1, \dots, b_n$. For convenience, we also define $b_0 \equiv 0$ and $b_{n+1} \equiv \infty$. The interval $[b_j, b_{j+1})$ is called the *region* for breakpoint j, or simply region j for short. Each region j, $j = 0, \dots, n$, is associated with a purchase price c_j. The costs are decreasing in j: $c_0 > c_1 > \cdots > c_n$. The total purchase cost, denoted $c(Q)$, is calculated in each of the discount structures as follows:

- *All-units discounts.* All units in the order incur the price determined by the breakpoint. That is, if $Q \in [b_j, b_{j+1})$, then the total purchase cost is $c(Q) = c_j Q$.

- *Incremental discounts.* The units in each region incur the purchase price for that region. That is, if $Q \in [b_j, b_{j+1})$, then the total purchase cost is

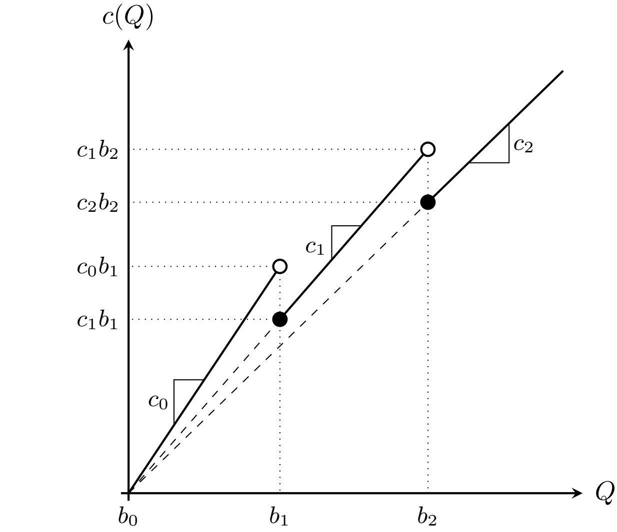
$$c(Q) = \sum_{i=0}^{j-1} c_i(b_{i+1} - b_i) + c_j(Q - b_j). \tag{3.17}$$

(Note that $c(Q)$ does not include the fixed ordering cost.) Figure 3.4 plots $c(Q)$ as a function of Q for both all-units and incremental discounts.

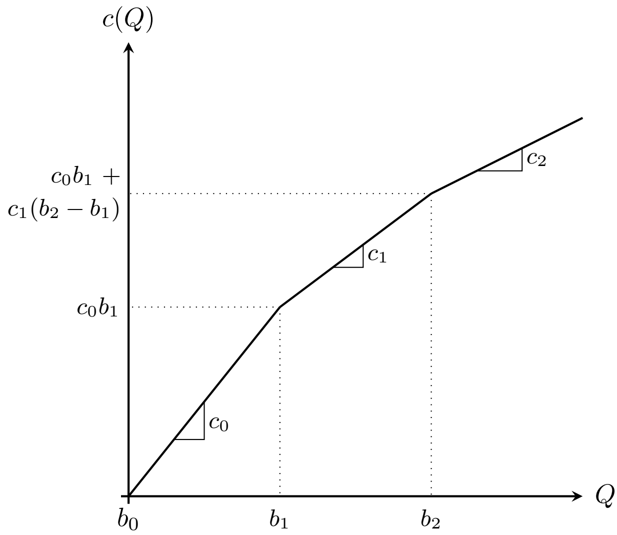
□ **EXAMPLE 3.5**

Suppose that Joe's candy supplier (from Example 3.1) charges \$0.75 per candy bar if Joe orders 0–400 candy bars, \$0.72 each for 401–800, and \$0.68 each for 800 or more. That is, $b_1 = 400$, $b_2 = 800$, $c_0 = 0.75$, $c_1 = 0.72$, and $c_2 = 0.68$. Figures 3.5(a) and 3.5(b) depict the total purchase cost, $c(Q)$, for the all-units and incremental discount structures, respectively. □

We will formulate models to determine the optimal order quantity under both discount structures. In both cases, the approach will amount to solving multiple EOQ problems, one for each region, and using their solutions to determine the solution to the original problem.



(a) All-units discounts.



(b) Incremental discounts.

Figure 3.4 Total purchase cost $c(Q)$ under quantity discounts.

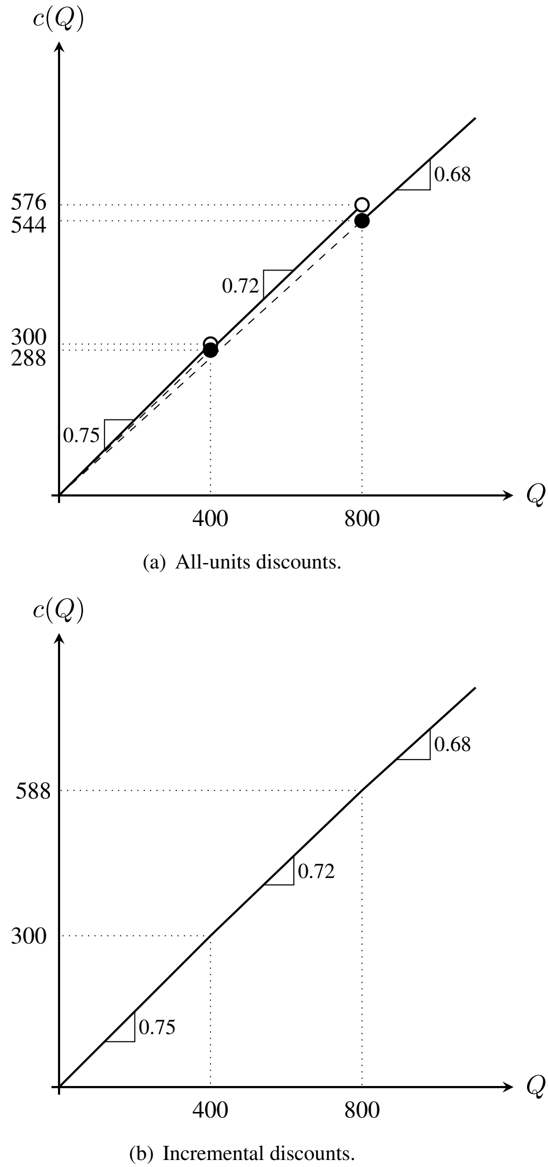


Figure 3.5 Total purchase cost $c(Q)$ for Example 3.5.

3.4.1 All-Units Discounts

We can no longer ignore the purchase cost as we did in (3.3). In fact, not only do we need to include the purchase cost itself, but we must also account for the fact that the holding cost typically depends on the purchase cost, as discussed in Section 3.1.3. Let i be the annual holding cost rate expressed as a percentage of the purchase cost. That is, if $i = 0.25$ and $c = 100$, then $h = 25$ per year.

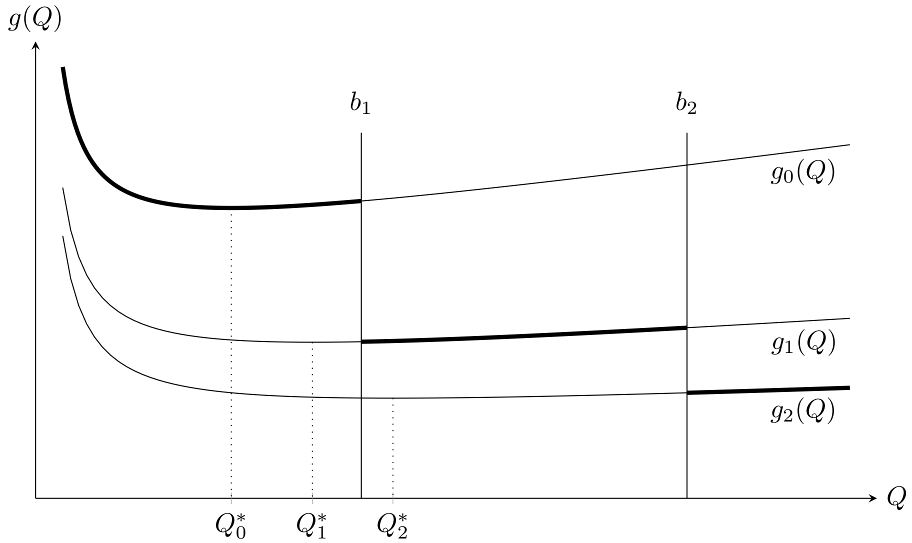


Figure 3.6 Total cost curves for all-units quantity discount structure.

Suppose we knew that the optimal order quantity lies in region j. Then we would simply need to find the Q that minimizes the EOQ cost function for region j:

$$g_j(Q) = c_j\lambda + \frac{K\lambda}{Q} + \frac{ic_jQ}{2}. \tag{3.18}$$

As j increases, c_j decreases, $g_j(Q)$ shifts down and becomes flatter, and its minimum point moves to the right; see Figure 3.6. The heavy segments of the cost curves identify the "active" cost function in each region. Our objective is to minimize $g(Q)$, the discontinuous function defined by the heavy segments.

The function $g_j(Q)$ has the same structure as $g(Q)$ in (3.3) except for the additional constant. Therefore, its minimizer is given by

$$Q_j^* = \sqrt{\frac{2K\lambda}{ic_j}}. \tag{3.19}$$

Of course, if Q_j^* falls outside of region j, then if the firm orders Q_j^*, it will incur a cost other than $g_j(Q_j^*)$. Q_j^* is meaningless in this case. We say that Q_j^* is *realizable* if it lies in region j. In Figure 3.6, only Q_0^* is realizable. Does this mean that Q_0^* is necessarily the optimal solution? No: The breakpoints to the right of Q_0^* are also candidates. The optimal order quantity always equals either the largest realizable Q_j^* or one of the breakpoints to its right. (Why?)

Therefore, we can determine Q^* as follows. First, we calculate Q_j^* for each j. Let Q_i^* be the largest realizable Q_j^*, and $g_i(Q_i^*)$ its cost. We then evaluate $g_j(b_j)$ for each b_j greater than Q_i^*. Finally, we set Q^* to the quantity with the lowest cost (Q_i^* if $g_i(Q_i^*)$ is the lowest cost, and b_j if $g_j(b_j)$ is the lowest cost for some j).

Since Q_j^* increases as j increases, if we start in region n when we calculate Q_j^* and work backward, we can stop as soon as we find one realizable Q_j^*; this is necessarily the largest realizable Q_j^*.

☐ **EXAMPLE 3.6**

Recall from Example 3.1 that $\lambda = 1300$, $K = 8$, and $i = 0.3$. If candy purchases follow the quantity discount structure in Example 3.5, what is Joe's optimal order quantity?

We first determine the largest realizable Q_j^* by working backward from segment 2:

$$Q_2^* = \sqrt{\frac{2 \cdot 8 \cdot 1300}{0.3 \cdot 0.68}} = 319.3$$
$$Q_1^* = \sqrt{\frac{2 \cdot 8 \cdot 1300}{0.3 \cdot 0.72}} = 310.3$$
$$Q_0^* = \sqrt{\frac{2 \cdot 8 \cdot 1300}{0.3 \cdot 0.75}} = 304.1$$

Only Q_0^* is realizable, and it has cost

$$0.75 \cdot 1300 + \sqrt{2 \cdot 8 \cdot 1300 \cdot 0.3 \cdot 0.75} = 1043.4.$$

Next, we calculate the cost of the breakpoints to the right of Q_0^*:

$$g_1(400) = 0.72 \cdot 1300 + \frac{8 \cdot 1300}{400} + \frac{0.3 \cdot 0.72 \cdot 400}{2} = 1005.2$$
$$g_2(800) = 0.68 \cdot 1300 + \frac{8 \cdot 1300}{800} + \frac{0.3 \cdot 0.68 \cdot 800}{2} = 978.6$$

Therefore, the optimal order quantity is $Q = 800$, which incurs a purchase cost of \$0.68 and a total annual cost of \$978.60. ☐

3.4.2 Incremental Discounts

We now turn our attention to incremental discounts. The total cost function for region j is given by

$$g_j(Q) = \frac{c(Q)}{Q}\lambda + \frac{K\lambda}{Q} + \frac{i\frac{c(Q)}{Q}Q}{2},$$

where $c(Q)$ is given by (3.17). Note that the purchase cost term is no longer a constant with respect to Q, even within a given segment: As Q increases, so does the number of "cheap" units, and the average cost per unit decreases.

We can rewrite $g_j(Q)$ as

$$\begin{aligned} g_j(Q) =& \frac{1}{Q}\left[\sum_{i=0}^{j-1} c_i(b_{i+1} - b_i) - c_j b_j\right]\lambda + c_j\lambda + \frac{K\lambda}{Q} \\ &+ \frac{i}{2}\left[\sum_{i=0}^{j-1} c_i(b_{i+1} - b_i) - c_j b_j\right] + \frac{ic_jQ}{2} \\ =& c_j\lambda + \frac{i\bar{c}_j}{2} + \frac{(K + \bar{c}_j)\lambda}{Q} + \frac{ic_jQ}{2}, \end{aligned} \tag{3.20}$$

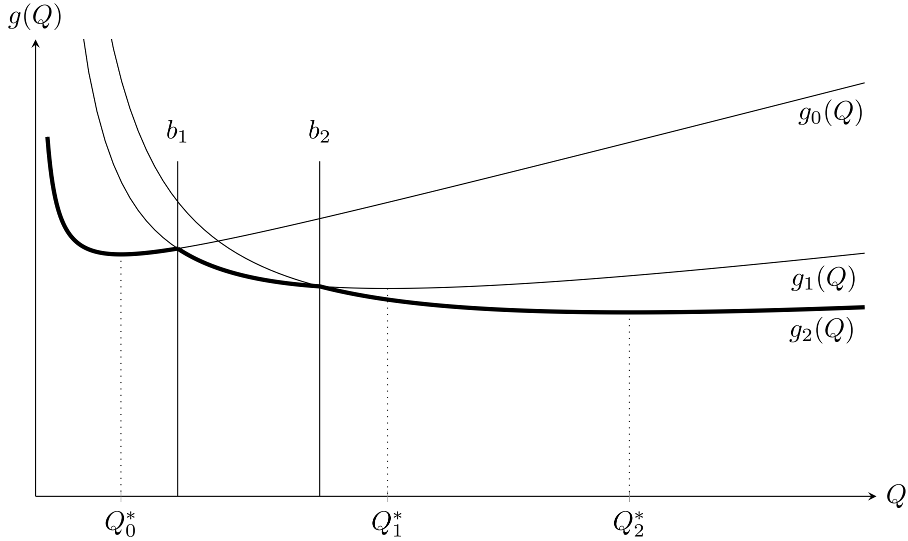


Figure 3.7 Total cost curves for incremental quantity discount structure.

where

$$\bar{c}_j = \sum_{i=0}^{j-1} c_i(b_{i+1} - b_i) - c_j b_j.$$

The right-hand side of (3.20) is structurally identical to the EOQ cost function; therefore, its minimizer is given by

$$Q_j^* = \sqrt{\frac{2(K + \bar{c}_j)\lambda}{ic_j}} \tag{3.21}$$

with cost

$$g_j(Q_j^*) = c_j\lambda + \frac{i\bar{c}_j}{2} + \sqrt{2(K + \bar{c}_j)\lambda i c_j}. \tag{3.22}$$

Figure 3.7 plots $g_j(Q)$ for a two-breakpoint problem. As a rule, $g_j(Q)$ is always the lowest curve in region j because the functions are convex and are equal at the breakpoints. On the other hand, Q_j^* is not always realizable. (In the figure, Q_1^* is not realizable.) Our objective is to minimize $g(Q)$, the continuous, piecewise function defined by the heavy segments.

If Q_j^* is not realizable, then clearly it cannot be optimal for $g(Q)$, and moreover, its breakpoints cannot be optimal either. (Why?) Therefore, the optimal order quantity is equal to the realizable Q_j^* that has the lowest cost.

☐ **EXAMPLE 3.7**

Return to Example 3.6 and suppose now that Joe faces an incremental quantity discount structure with the same breakpoints and purchase costs. What is Joe's optimal order quantity?

We first determine $\bar{c}_j$ for each j:

$$\bar{c}_0 = 0$$

$$\bar{c}_1 = 0.75 \cdot 400 - 0.72 \cdot 400 = 12$$
$$\bar{c}_2 = 0.75 \cdot 400 + 0.72 \cdot 400 - 0.68 \cdot 800 = 44$$

Next, we calculate Q_j^* for each j:

$$Q_0^* = \sqrt{\frac{2(8+0)1300}{0.3 \cdot 0.75}} = 304.1$$
$$Q_1^* = \sqrt{\frac{2(8+12)1300}{0.3 \cdot 0.72}} = 490.7$$
$$Q_2^* = \sqrt{\frac{2(8+44)1300}{0.3 \cdot 0.68}} = 814.1$$

All three solutions are realizable. Using (3.22), these solutions have the following costs:

$$g_0(Q_0^*) = 0.75 \cdot 1300 + \frac{0.3 \cdot 0}{2} + \sqrt{2(8+0)1300 \cdot 0.3 \cdot 0.75} = 1043.4$$
$$g_1(Q_1^*) = 0.72 \cdot 1300 + \frac{0.3 \cdot 12}{2} + \sqrt{2(8+12)1300 \cdot 0.3 \cdot 0.72} = 1043.8$$
$$g_2(Q_2^*) = 0.68 \cdot 1300 + \frac{0.3 \cdot 44}{2} + \sqrt{2(8+44)1300 \cdot 0.3 \cdot 0.68} = 1056.7$$

Therefore, the optimal order quantity is $Q = 304.1$, which incurs a total annual cost of \$1043.40. □

3.4.3 Modified All-Units Discounts

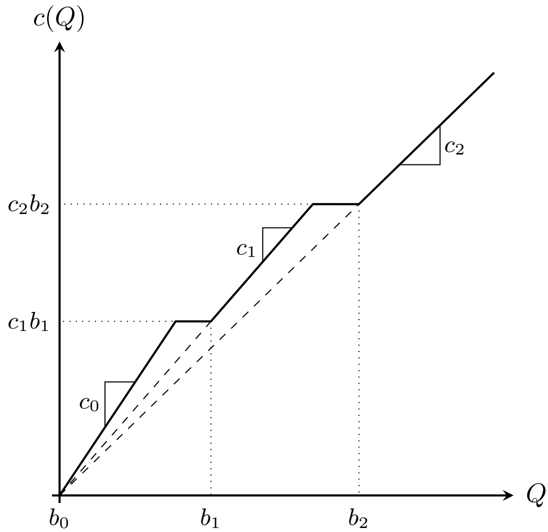
All-units discounts are somewhat problematic because, for order quantities Q just to the left of breakpoint j, it is cheaper to order b_j than to order Q, even though $Q < b_j$. For example, under the cost structure in Example 3.5, it costs \$292.50 to purchase 390 units but \$288.00 to purchase 400 units. (See Figure 3.5(a).)

In practice, suppliers usually allow the buying firm to pay the lower price—\$288.00 in the example above—for order quantities that fall into this awkward zone. This is especially true for transportation costs, since all-units discounts are common in shipping, with the cost determined based on the weight shipped. If a shipment totals, say, 390 kg but it is cheaper to ship 400 kg, the firm could add 10 kg worth of bricks to the shipment, but a solution that is preferable for both the shipper and the transportation company is for the firm to "ship x, declare y"—for example, ship 390 kg, declare 400 kg.

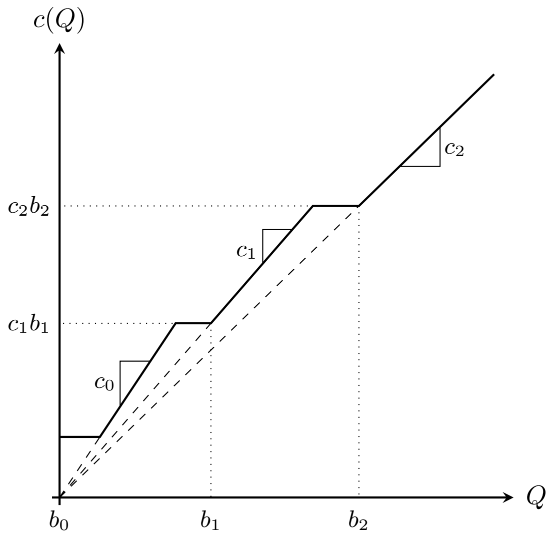
This structure is sometimes known as the *modified all-units discount structure*. Its $c(Q)$ curve is displayed in Figure 3.8(a). The flat portions of the curve represent the regions in which the firm orders or ships one quantity but declares a greater quantity.

Sometimes, there is also a minimum charge for each order or shipment, in which case there is an additional horizontal segment at the start of the $c(Q)$ curve; see Figure 3.8(b).

A special case of the modified all-units discount structure is the *carload discount structure*, in which the b_j are equally spaced and c_j is the same for all j. This structure arises from rail or truck carload shipments, in which the transportation company charges a per-unit cost c for each unit shipped, up to some maximum cost for each car. Once the capacity of a car is exceeded, a new car begins, at a cost of c per unit, and so on.



(a) Without minimum cost.



(b) With minimum cost.

Figure 3.8 Total purchase cost $c(Q)$ for modified all-units discounts structure.

Unfortunately, modified all-units discount structures are much more difficult to analyze than the discount structures discussed above. (See, for example, Chan et al. (2002).) We omit further discussion here.

3.5 THE EOQ WITH PLANNED BACKORDERS

We assumed in Section 3.2.1 that backorders are not allowed. In this section, we discuss a variant of the EOQ problem in which backorders are allowed. Since demand is determin-

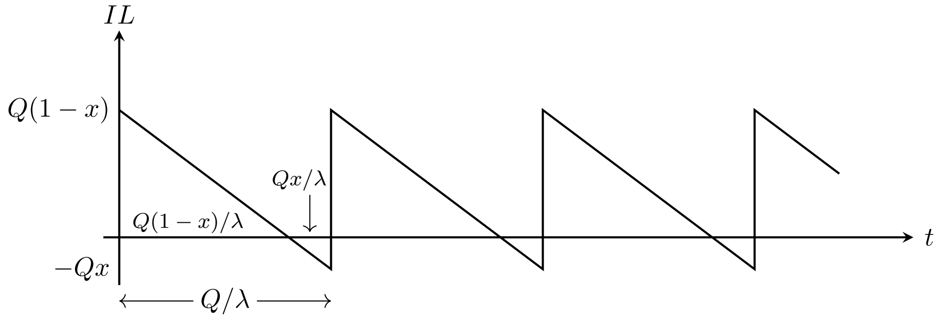


Figure 3.9 EOQB inventory curve.

istic, we have the same number of backorders in every order cycle—they are "planned" backorders. (See Figure 3.9.) We'll call this model the *EOQ with backorders* (EOQB).

Let p be the backorder penalty per item per year, and let x be the fraction of demand that is backordered. Both Q and x are decision variables. The holding cost is charged based on on-hand inventory; the average on-hand inventory is given by

$$\frac{Q(1-x)^2}{2}.$$

Similarly, the backorder cost is charged based on the number of backorders; the average backorder level is given by

$$\frac{Qx^2}{2}.$$

(Compute the area under the triangle, then divide by the length of an order cycle.) Finally, the number of orders per year is given by λ/Q, just like in the EOQ model.

Therefore, the total average cost per year in the EOQB is given by

$$g(Q,x) = \frac{hQ(1-x)^2}{2} + \frac{pQx^2}{2} + \frac{K\lambda}{Q}. \tag{3.23}$$

Note that g is a function of both Q and x. Therefore, to minimize it, we need to take partial derivatives with respect to both variables and set them equal to 0.

$$\frac{\partial g}{\partial x} = -hQ(1-x) + pQx = 0 \tag{3.24}$$

$$\frac{\partial g}{\partial Q} = \frac{h(1-x)^2}{2} + \frac{px^2}{2} - \frac{K\lambda}{Q^2} = 0 \tag{3.25}$$

Let's first look at (3.24):

$$\begin{aligned} -hQ(1-x) + pQx &= 0 \\ \iff h(1-x) &= px \\ \iff x^* &= \frac{h}{h+p} \end{aligned} \tag{3.26}$$

Interestingly, x^* does not depend on Q; even if we choose a suboptimal Q, the optimal x to choose is still $h/(h+p)$. At this point, we could substitute $h/(h+p)$ for x in (3.25) and solve for Q, but instead we'll plug x^* into $g(Q, x)$:

$$\begin{aligned} g(Q, x^*) &= \frac{hQ}{2}\left(\frac{p}{h+p}\right)^2 + \frac{pQ}{2}\left(\frac{h}{h+p}\right)^2 + \frac{K\lambda}{Q} \\ &= \frac{Q}{2}\left(\frac{p^2h + h^2p}{(h+p)^2}\right) + \frac{K\lambda}{Q} \\ &= \frac{hp}{h+p}\frac{Q}{2} + \frac{K\lambda}{Q} \end{aligned}$$

This is exactly the same form as the EOQ cost function (3.3) with the holding cost h replaced by $hp/(h+p)$. In other words, the EOQB cost function (assuming x is set optimally) is equivalent to the EOQ cost function with the holding cost h scaled by $p/(h+p)$. Therefore we can use (3.6) and (3.7) to obtain the optimal Q and the optimal cost for the EOQB, as stated in the next theorem.

Theorem 3.5 *In the EOQ model with backorders, the optimal solution and cost are given by*

$$Q^* = \sqrt{\frac{2K\lambda(h+p)}{hp}} \tag{3.27}$$

$$x^* = \frac{h}{h+p} \tag{3.28}$$

$$g(Q^*, x^*) = \sqrt{\frac{2K\lambda hp}{h+p}} \tag{3.29}$$

How do the optimal solution and cost in Theorem 3.5 compare to the analogous quantities from the EOQ model? First, comparing (3.29) and (3.7), we can see that the optimal cost is smaller in the EOQB than in the EOQ. This makes sense, since the EOQ is a special case of the EOQB in which the constraint $x = 0$ has been added. From (3.27), we can see that the optimal order quantity is greater in the EOQB than in the EOQ. This is because placing larger orders in the EOQB does not require us to carry quite as much inventory as it does in the EOQ, and therefore, the extra flexibility offered by the backorder option allows us to place larger orders.

As $p \to \infty$, Q^* approaches the optimal EOQ order quantity, x^* approaches 0, and the optimal cost approaches the EOQ optimal cost.

Note also that x is strictly greater than 0, provided that h is. Therefore, it is *always* optimal to allow some backorders. To see why, suppose we set $x = 0$—then the EOQB inventory curve in Figure 3.9 collapses to the EOQ curve in Figure 3.2. Now, if we increase x slightly, we create a tiny negative triangle at the end of each cycle in Figure 3.9, incurring a tiny backorder cost. (See Figure 3.10.) But we also reduce the height of the positive part of the inventory curve throughout the rest of the cycle, resulting in a substantial savings in holding cost. As we continue to increase the number of backorders, the marginal savings in holding cost decreases and the marginal increase in backorder cost increases. At some point, the marginal cost of adding a backorder will outweigh the marginal savings in holding cost, so we will have an x^* somewhere between 0 and 1.

What if we consider the same model but assume that unmet demands are lost, rather than backordered? It turns out that in this case, it is optimal either to meet every demand ($x = 0$) or to meet no demands ($x = 1$)—see Problem 3.16.

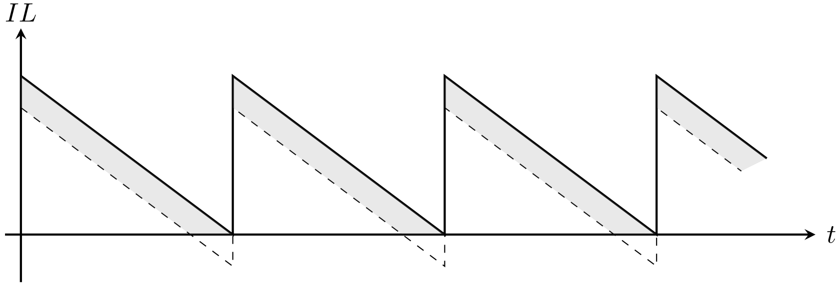


Figure 3.10 Inventory–backorder trade-off in EOQB.

☐ **EXAMPLE 3.8**

Recall Example 3.1. Suppose Joe is willing to stock out occasionally and estimates that each backorder costs the store \$5 in lost profit and loss of good will. What is the optimal order quantity, the optimal fill rate (fraction of demand met from stock), and the optimal cost?

$$Q^* = \sqrt{\frac{2K\lambda(h+p)}{hp}} = \sqrt{\frac{2 \cdot 8 \cdot 1300(0.225+5)}{0.225 \cdot 5}} = 310.81$$

$$x^* = \frac{h}{h+p} = 0.0431$$

$$g(Q^*, x^*) = \sqrt{\frac{2K\lambda hp}{h+p}} = \sqrt{\frac{2 \cdot 8 \cdot 1300 \cdot 0.225 \cdot 5}{0.225+5}} = 66.92$$

The fill rate is $1 - x^* = 0.9569$. The cost has decreased by 2.2% versus the cost without backorders. ☐

3.6 THE ECONOMIC PRODUCTION QUANTITY MODEL

In a manufacturing environment, the amount of time required to produce a batch of items usually depends on how large the batch is—producing more items requires more time. The EOQ model cannot handle this feature, since it assumes that orders are received after a deterministic (possibly zero) lead time, regardless of the order quantity. In other words, the EOQ assumes that the production rate is infinite—an arbitrary number of items can be produced in a fixed amount of time. This assumption may be reasonable in settings in which the firm is placing orders to an outside supplier that holds finished goods in inventory, or whose capacity is much larger than the firm's order quantity, so that the production time is negligible. In this section, we discuss a variant of the EOQ model that allows the production rate to be finite and is therefore more applicable to manufacturing settings. It is known as the *economic production quantity* (EPQ) model. The EPQ was introduced by Taft (1918, as cited by Erlenkotter (1990)). It is sometimes known as the economic production lot (EPL) problem.

Let μ be the production rate, i.e., the firm can produce μ items per year. We assume $\mu > \lambda$ (otherwise the manufacturing process cannot keep up with the demand). The

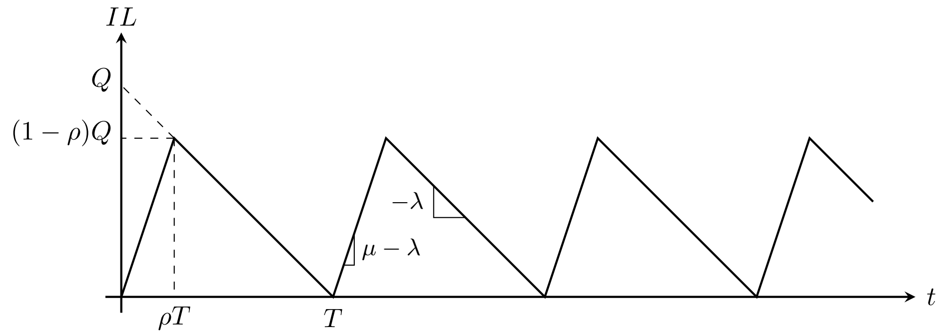


Figure 3.11 EPQ inventory level curve.

manufacturing process is active during a portion of the time (called *active intervals*) and is idle otherwise, and during active intervals, the process adds finished goods to inventory at a rate of μ. Meanwhile, the demand process is ongoing, reducing the inventory at a rate of λ. Let $\rho = \lambda/\mu$ be the *utilization ratio*, which indicates the portion of time the system is active. Q is now interpreted as a production batch size rather than an order quantity.

The process is depicted in Figure 3.11. Note that during active intervals, the inventory increases at a rate $\mu - \lambda$ since items are being added to inventory by the manufacturing process and withdrawn from it by the demand process simultaneously. Since we still initiate the replenishment process after exactly Q items have been demanded, the order interval T still equals Q/λ years. Moreover, since we produce exactly Q units in an active interval, the active interval must last $Q/\mu = \rho T$ years. This means that the maximum inventory level, which occurs ρT years into each cycle, is $\rho T(\mu - \lambda) = (1-\rho)Q$.

The fixed cost per year is still $K\lambda/Q$, as in the EOQ model, since $T = Q/\lambda$. The average inventory level is $(1-\rho)Q/2$, so the average annual holding cost is $h(1-\rho)Q/2$. Therefore, the total annual cost is

$$g(Q) = \frac{K\lambda}{Q} + \frac{h(1-\rho)Q}{2}. \tag{3.30}$$

We could find the Q that minimizes this cost function by differentiating, as we did for the EOQ, but it is simpler to recognize that (3.30) differs from (3.3) only by the constant $(1-\rho)$ in the second term. In other words, the EPQ is equivalent to the EOQ with the holding cost parameter h scaled by $1-\rho$. Therefore, the optimal solution to the EPQ, and its cost, are as given in the next theorem.

Theorem 3.6 *In the EPQ model, the optimal solution and cost are given by*

$$Q^* = \sqrt{\frac{2K\lambda}{h(1-\rho)}} \tag{3.31}$$

$$g(Q^*) = \sqrt{2K\lambda h(1-\rho)}. \tag{3.32}$$

Proof. Follows from replacing h with $h(1-\rho)$ in Theorem 3.1. ■

Since $\rho < 1$, the optimal EPQ solution is larger than that of the EOQ, while the optimal EOQ cost is smaller. Both results are justified by the fact that items arrive later after the replenishment order in the EPQ than they do in the EOQ, and therefore, the holding cost for a given Q is smaller. Note also that as $\mu \to \infty$, the EPQ reduces to the EOQ.

3.7 PERIODIC REVIEW: THE WAGNER–WHITIN MODEL

3.7.1 Problem Statement

We now shift our attention to a periodic-review model known as the *Wagner–Whitin model* (Wagner and Whitin 1958). Similar to the EOQ model, the Wagner–Whitin model assumes that the demand is deterministic, there is a fixed cost to place an order, and stock-outs are not allowed. The objective is to choose order quantities to minimize the total cost. However, unlike the EOQ model, the Wagner–Whitin model allows the demand to change over time—to be different in each period. This model is sometimes referred to as the dynamic economic lot-sizing (DEL) model or the uncapacitated lot-sizing (ULS) model.

Because of the fixed cost, it may not be optimal to place an order in every time period. However, we will show that, as in the EOQ, optimal solutions have the zero-inventory ordering (ZIO) property. Therefore, the problem boils down to deciding how many whole periods' worth of demand to order at once.

Unlike the infinite-horizon EOQ model, the Wagner–Whitin model considers a finite horizon, consisting of T periods. In each period, we must decide whether to place a replenishment order, and if so, how large an order to place. The demand in period t is given by d_t, and stockouts are not allowed. The lead time is 0. As in the EOQ model, there is a fixed cost K per order and an inventory holding cost h per unit per period. (Note that h represents the holding cost per year in the EOQ model but per period here.) One could also include a purchase cost c, but since the total number of units ordered throughout the horizon is constant (independent of the ordering pattern), it is safe to ignore this cost.

Assume that the on-hand inventory is 0 at time 0. In each time period, the following events occur, in the following order:

1. The replenishment order, if any, is placed and is received instantly.

2. Demand occurs and is satisfied from inventory.

3. Holding costs are assessed based on the on-hand inventory.

(This type of timeline is known as a *sequence of events*. It is important to specify the sequence of events clearly in periodic-review models. For example, the holding costs would be very different if events 2 and 3 were reversed.)

We first formulate this model as a mixed-integer optimization problem (MIP). We will then discuss a dynamic programming (DP) algorithm for solving it.

3.7.2 MIP Formulation

Our formulation will use the following decision variables:

q_t = the number of units ordered in period t
y_t = 1 if we order in period t, 0 otherwise
x_t = the inventory level at the end of period t

We also define $x_0 \equiv 0$. Then the Wagner–Whitin model can be formulated as follows:

$$\text{minimize} \quad \sum_{t=1}^{T} (Ky_t + hx_t) \tag{3.33}$$

$$\text{subject to} \quad x_t = x_{t-1} + q_t - d_t \quad \forall t = 1, \ldots, T \tag{3.34}$$

$$q_t \leq My_t \quad \forall t = 1, \ldots, T \tag{3.35}$$

$$x_t \geq 0 \quad \forall t = 1, \ldots, T \tag{3.36}$$

$$q_t \geq 0 \quad \forall t = 1, \ldots, T \tag{3.37}$$

$$y_t \in \{0, 1\} \quad \forall t = 1, \ldots, T \tag{3.38}$$

The objective function (3.33) calculates the fixed cost (for each period in which we place an order) plus the cost of holding inventory at the end of each period. Constraints (3.34) are the *inventory-balance constraints*: They say that the ending inventory in period t is equal to the starting inventory, plus the new units ordered, minus the demand. Constraints (3.35) prohibit q_t from being positive unless y_t is 1. Here, M is a large number; it could be set to $\sum_{s=t}^{T} d_s$, for example. Constraints (3.36)–(3.37) are nonnegativity constraints. In particular, (3.36) also prohibits stockouts by requiring every period to end with nonnegative inventory. Finally, constraints (3.38) are integrality constraints on the y variables.

This problem can be interpreted as a simple supply chain network design problem (to be more precise, an arc design problem; see Section 8.7.2). It can be solved as an MIP, but it is more common to solve it using DP or as a shortest path problem, as we discuss in the next section. See Pochet and Wolsey (1995, 2006) for thorough discussions of mathematical programming formulations for this and other lot-sizing models. See also Case Study 3.1 for an alternate formulation approach for a similar problem.

3.7.3 Dynamic Programming Algorithm

The DP algorithm depends on the following result:

Theorem 3.7 *Every optimal solution to the Wagner–Whitin model has the ZIO property; that is, it is optimal to place orders only in time periods in which the initial inventory is zero.*

Proof. Suppose (for a contradiction) there is an optimal solution in which an order is placed in period t even though the inventory level at the beginning of period t is positive; i.e., $x_{t-1} > 0$. The x_{t-1} units in inventory were ordered in a period before t and incurred a holding cost to be held from period $t-1$ to t. If these items had instead been ordered in period t, then (1) the holding cost would decrease since fewer units are held in inventory, and (2) the fixed cost would stay the same since the number of orders would not change, only the size of each order. This contradicts the assumption that the original policy is optimal; hence, every optimal solution must have the ZIO property. ■

Theorem 3.7 and its proof assume that $h > 0$; if h may equal 0, then the theorem would read "There exists an optimal solution..."

As a corollary to Theorem 3.7, each order is of a size equal to the total demand in an integer number of subsequent periods; that is, in period t we either order d_t, or $d_t + d_{t+1}$, or $d_t + d_{t+1} + d_{t+2}$, and so on. The problem then boils down to deciding in which periods to order. We formulate this problem as a DP.

Let θ_t be the optimal cost in periods $t, t+1, \ldots, T$ if we place an order in period t (and act optimally thereafter). We can define θ_t recursively in terms of θ_s for later periods s. First define $\theta_{T+1} \equiv 0$. Then

$$\theta_t = \min_{t < s \leq T+1} \left\{ K + h \sum_{i=t}^{s-1} (i-t) d_i + \theta_s \right\}. \tag{3.39}$$

The minimization determines the next period s in which we will place an order, assuming that we order in period t. (Setting $s = T+1$ means we never order again; the order in period t is the last order.) A given choice of s is evaluated using the expression inside the braces. The first two terms calculate the cost incurred in periods t through $s-1$: the order cost of K, plus the holding cost for the items that will be held until future periods. (The d_t units demanded in period t will be held for 0 periods; d_{t+1} units will be held for 1 period; $\ldots$; and d_{s-1} units will be held for $s-1-t$ periods.) A new order will be placed in period s, and θ_s includes the cost in period s and all future periods.

The DP algorithm for the Wagner–Whitin problem is summarized in Algorithm 3.1. At the conclusion of the algorithm, θ_1 equals the cost of the optimal solution. The optimal solution itself is obtained by "backtracking"—we place orders in period 1, period $s(1)$, period $s(s(1))$, and so on.

Algorithm 3.1 Wagner–Whitin algorithm

```
1: θ_{T+1} ← 0                                        ▷ Initialization
2: for t = T, ..., 1 do                               ▷ Main loop
3:     θ_t ← right-hand side of (3.39)                ▷ Minimization over s
4:     s(t) ← argmin in right-hand side of (3.39)
5: end for
6: return θ_t, s(t) for all t = 1, ..., T
```

The complexity of the algorithm is $O(T^2)$ since step 2 requires $O(T)$ operations and must be performed $O(T)$ times. Faster algorithms, which run in $O(T)$ time, have been developed for this problem but will not be discussed here (Federgruen and Tzur 1991, Wagelmans et al. 1992). Despite the efficiency of this algorithm, a number of heuristics have been introduced and are still popular in practice. These include Silver–Meal, part period balancing, least unit cost, and other heuristics (Silver et al. 1998). One explanation for the persistent use of these approximate methods is that they tend to be less sensitive to changes in the data, so that as demand forecasts change for several periods into the future, the current production plan doesn't change much.

The Wagner–Whitin model can equivalently be represented by a network with $T+1$ nodes in which each node represents a time period and an arc from period t to period s represents ordering in period t to satisfy the demands of periods $t, t+1, \ldots, s-1$. The cost of this arc is

$$K + h \sum_{i=t}^{s-1} (i-t) d_i. \tag{3.40}$$

Solving the Wagner–Whitin problem is equivalent to finding a shortest path through this network (which is, in turn, equivalent to solving the DP given above). Figure 3.12 depicts the network for a 4-period problem. Note that there is one extra node, node 5, called the

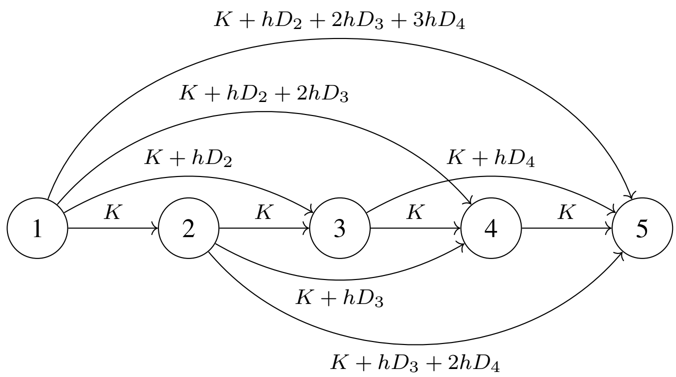


Figure 3.12 Wagner–Whitin network.

"dummy node," that serves as a sink for arcs representing ordering from the current time period until the end of the horizon.

☐ **EXAMPLE 3.9**

A garden center sells bags of organic compost for vegetable gardens. Compost is heavy, and special trucks must be used to transport it, so shipping is expensive; each order therefore incurs a fixed cost of \$500. The holding cost for each cubic meter of compost is \$2 per period. We consider a 4-period planning horizon. The demand for compost in periods 1–4 is 90, 120, 80, and 70 cubic meters, respectively. Find the optimal order quantity in each period and the total cost.

From (3.39), we have the following:

$$
\begin{aligned}
\theta_5 =& 0 \\
\theta_4 =& K + h(0 \cdot d_4) + \theta_5 \\
=& 500 \quad [s(4) = 5] \\
\theta_3 =& \min\{K + h(0 \cdot d_3) + \theta_4, K + h(0 \cdot d_3 + 1 \cdot d_4) + \theta_5\} \\
=& \min\{1000, 640\} \\
=& 640 \quad [s(3) = 5] \\
\theta_2 =& \min\{K + h(0 \cdot d_2) + \theta_3, K + h(0 \cdot d_2 + 1 \cdot d_3) + \theta_4, \\
& K + h(0 \cdot d_2 + 1 \cdot d_3 + 2 \cdot d_4) + \theta_5\} \\
=& \min\{1140, 1160, 940\} \\
=& 940 \quad [s(2) = 5] \\
\theta_1 =& \min\{K + h(0 \cdot d_1) + \theta_2, K + h(0 \cdot d_1 + 1 \cdot d_2) + \theta_3, \\
& K + h(0 \cdot d_1 + 1 \cdot d_2 + 2 \cdot d_3) + \theta_4, \\
& K + h(0 \cdot d_1 + 1 \cdot d_2 + 2 \cdot d_3 + 3 \cdot d_4) + \theta_5\} \\
=& \min\{1440, 1380, 1560, 1480\} \\
=& 1380 \quad [s(1) = 3]
\end{aligned}
$$

Therefore, we order in periods 1 and $s(1) = 3$; the optimal order quantities are $Q_1 = d_1 + d_2 = 210$ and $Q_3 = d_3 + d_4 = 150$ cubic meters; and the total cost is 1380. □

3.7.4 Extensions

Many of the assumptions made in Section 3.7.1 can be relaxed without making the problem substantially harder. For example, period-specific costs (h_t, K_t, c_t) can easily be accommodated. Similarly, nonzero lead times can be handled, provided the lead time is still fixed and constant. Positive initial inventories can be handled with appropriate modifications to the cost function in period 1.

Other extensions are considerably more difficult. For example, we assumed implicitly that there were no capacity constraints—an order can be placed of any size, and any amount of inventory can be carried over. Capacitated versions of the Wagner–Whitin model turn out to be NP-hard (Florian et al. 1980). Backlogging and concave order costs (instead of linear) are considered by Zangwill (1966); the model is still polynomially solvable, but the solution approach is less tractable than the DP presented here.

CASE STUDY 3.1 Ice Cream Production and Inventory at Scotsburn Dairy Group

Scotsburn Dairy Group is one of Canada's largest producers of ice cream and other dairy products. Its factory in Truro, Nova Scotia produces nearly 30 million liters of ice cream per year. Scotsburn collaborated with the industrial engineering department at Dalhousie University to optimize the production and inventory of ice cream at the Truro facility. The collaboration first began as an undergraduate design project, then a Master's project. The approach is described by Gunn et al. (2014).

The team developed a hierarchical planning process that includes a monthly model for setting inventory targets and staffing levels over a 1-year horizon; a weekly model to determine how much of each stock-keeping unit (SKU) to produce per week; and a daily model to optimize the production schedule. All three were formulated as integer programming (IP) models. We focus on the weekly model, which is an extension of the Wagner–Whitin model discussed in Section 3.7.

The Truro facility produces over 300 SKUs of ice cream, which the researchers aggregated into just over 100 product families. The weekly model determines how much of each family to produce in each week over a 13-week horizon. The model is used on a rolling-horizon basis, meaning that the company only implements next week's plan; it then solves the model again for another 13-week horizon.

Let F be the set of product families. Let a_t^+ and a_t^- be the maximum and minimum number of production hours that may be used in week t, respectively. (These are outputs from the monthly planning model.) Let u_{f,t_1,t_2} be the number of production hours required to produce family $f \in F$ in week t_1 to cover the demand in weeks $t_1, \ldots, t_2$, and let c_{f,t_1,t_2} be the cost (including both fixed and holding costs) to do so. Similar to (3.40),

$$c_{f,t_1,t_2} = K_f + h_f \sum_{t=t_1}^{t_2} (t - t_1) d_{tf},$$

where the parameters are as in Section 3.7 but are now also indexed by the product family, f. The decision variable x_{f,t_1,t_2} equals 1 if family f is produced in week t_1 in order to cover the demand in weeks $t_1, \ldots, t_2$, and 0 otherwise. Note that this is a different type of formulation than that used in Section 3.7.2 since the decision variables determine how many periods' of demand to produce rather than modeling the production and inventory levels explicitly.

The Scotsburn weekly model can be formulated as follows[2]:

$$\text{minimize} \quad \sum_{f \in F} \sum_{t_1=1}^{T} \sum_{t_2=t_1}^{T} c_{f,t_1,t_2} x_{f,t_1,t_2} \tag{3.41}$$

$$\text{subject to} \quad \sum_{t_1=1}^{t} \sum_{t_2=t}^{T} x_{f,t_1,t_2} = 1 \qquad \forall f \in F, \forall t = 1, \ldots, T \tag{3.42}$$

$$a_t^- \le \sum_{f \in F} \sum_{t_2 \ge t} u_{f,t,t_2} x_{f,t,t_2} \le a_t^+ \qquad \forall t = 1, \ldots, T \tag{3.43}$$

$$x_{f,t_1,t_2} \in \{0,1\} \qquad \forall f \in F, \forall t_1, t_2 = 1, \ldots, T \tag{3.44}$$

The objective function (3.41) calculates the total production and inventory costs. Constraints (3.42) ensure that the demand for each product family f in each week t is produced in some production run that includes period t. Constraints (3.43) require the total number of production hours used in period t to be within the allowable range. Constraints (3.44) are integrality constraints.

Scotsburn solves this model using CPLEX, which can solve a typical instance—roughly 10,000 variables and 2,000 constraints—to 2% optimality within a few minutes. The company reports that the full project—including the monthly, weekly, and daily planning models—helped to improve the fill rate (fraction of demand met from stock) from 90.2% to 96.2%; it also improved the production rate (units produced per hour) by 3% as a result of having fewer time-consuming production setups.

PROBLEMS

3.1 (EOQ for Steel) An auto manufacturer uses 500 tons of steel per day. The company pays \$1100 per ton of steel purchased, and each order incurs a fixed cost of \$2250. The holding cost is \$275 per ton of steel per year. Using the EOQ model, calculate the optimal order quantity, cycle length, and average cost per year.

3.2 (EOQ for MP3s) Suppose that your favorite electronics store maintains an inventory of a certain brand and model of MP3 player. The store pays the manufacturer \$165 for each MP3 player ordered. Each order incurs a fixed cost of \$40 in order processing, shipping, etc. and requires a 2-week lead time. The store estimates that its cost of capital is 17% per year, and it estimates its other holding costs (warehouse space, insurance, etc.) at \$1 per MP3 player per month. The demand for MP3 players is steady at 40 per week.

[2]The real model includes multiple production lines and allows for overtime, but we omit these aspects for the sake of simplicity and instead assume that the factory has a single production line with hard constraints on the production hours available.

a) Using the EOQ model, calculate the optimal order quantity, reorder point (r), and average cost per year.

b) Now suppose that backorders are allowed, and that each backorder incurs a stockout penalty of \$60 per stockout per year. Using the EOQ model with planned backorders, calculate the optimal order quantity, stockout percentage (x), reorder point (r), and average cost per year. How much money would the store save per year by allowing stockouts, expressed as a percentage?

3.3 (EOQ for Cat Toys) Mason's Meows is a company that makes cat toys. The company sells 1200 toys per year. The firm incurs a fixed cost of \$150 in labor each time it starts up the manufacturing process to begin a new batch of toys. Each toy costs Mason's Meows \$9 to produce. The company's accountant recommends using a holding cost equal to 20% of the cost of the toy, per year.

a) What is the optimal batch size, Q^*? If the company uses batches of size Q^*, how many times per year, on average, will it start up the manufacturing process?

b) After careful analysis, the inventory team at Mason's Meows realized that the per-unit production cost is smaller if the batch size is larger. In particular, the production cost is \$9 per unit for batches of fewer than 400 units and \$7.50 per unit for batches of 400 or more units. Now what is the optimal batch size?

3.4 (EOQ for Vaccines) A medical clinic dispenses vaccines at a steady rate of 520 doses per month. Each order placed to the vaccine manufacturer incurs a fixed cost of \$140. Each vaccine dose held in inventory incurs a holding cost of \$3 per year.

a) Using the EOQ model, calculate the optimal order quantity, Q^*, and the optimal average cost per year, $g(Q^*)$.

b) Suppose that the fixed cost K increases. Will Q^* increase, decrease, or stay the same? Briefly explain your answer.

3.5 (EOQ for Automobile Components) An automobile manufacturing plant uses exactly 8 power-lock mechanisms per hour. Each replenishment order to the supplier of the power-lock mechanisms incurs a fixed cost of \$85. Each mechanism stored in inventory incurs a holding cost of \$1.50 per week.

a) Using the EOQ model, calculate the optimal order quantity, Q^*, and the optimal average cost per year, $g(Q^*)$.

b) Suppose that the plant must order in power-of-two multiples of 1 week. (That is, the plant can place an order every week, or every 2 weeks, or every 4 weeks, ..., or every $\frac{1}{2}$ week, or every $\frac{1}{4}$ week, ...) What is the optimal power-of-two order interval, and what is the cost ratio versus the optimal (non-power-of-two) solution?

3.6 (Snack Bar Inventory Management, Part 1) A snack bar at a certain theme park sees a (constant, deterministic, continuous) demand of 150 cases per day. (We are aggregating the various products sold by the snack bar into a single product and expressing its demand in terms of number of cases.) Replenishment orders are placed to a central warehouse located within the theme park, with negligible lead time, and it costs \$10 in labor costs to deliver an order to the snack bar from the warehouse. It costs \$1.20 per case per day in refrigeration costs and other holding costs to hold cases of food in inventory at the snack bar.

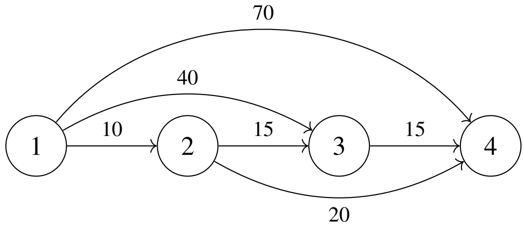


Figure 3.13 Shortest path network for Problem 3.7.

a) Calculate the optimal order quantity, Q^*, for the snack bar.

b) If the snack bar uses Q^* as its order quantity, how often will it order?

c) Suppose the snack bar must order in multiples of 20 cases. (That is, it must order 20 cases, or 40 cases, or 60 cases, or . . . .) Do you think the snack bar's costs will increase significantly due to this restriction? Briefly explain your answer.

3.7 (Snack Bar Inventory Management, Part 2) For the snack bar in Problem 3.6, suppose now that the demand is different on different days of the week, as given in the following table. Replenishment orders can only be placed at the start of each day. The fixed and holding costs are as given in Problem 3.6.

Day (#)	Day (Name)	Demand
1	Sunday	220
2	Monday	155
3	Tuesday	105
4	Wednesday	90
5	Thursday	170
6	Friday	210
7	Saturday	290

a) Assume that the snack bar uses a 7-day planning horizon, beginning on Sunday. Let c_{ts} be the cost to place an order on day t that will last through the end of day $s-1$, including both the fixed ordering cost and the holding cost. Calculate c_{12}, c_{47}, and c_{68}.

b) Suppose instead that the snack bar uses a 3-day planning horizon and that the shortest path network representing fixed and holding costs is as given in Figure 3.13. (The numbers in this figure come from different data than those in part (a).) On which day(s) should the snack bar place orders?

3.8 (EOQ with Nonzero Lead Time) Consider the EOQ model with fixed lead time $L > 0$ (Section 3.2.5). Prove that the average amount of inventory on order is equal to the lead-time demand.

3.9 (Change in Optimal EOQ Cost) Suppose we have two instances of the EOQ problem, h_1, K_1, λ_1 and h_2, K_2, λ_2, such that $\sqrt{2K_1\lambda_1 h_1} < \sqrt{2K_2\lambda_2 h_2}$. True, false, or indeterminate: The holding cost component (i.e., the $hQ/2$ part) of the optimal objective function value is greater under instance 2 than under instance 1. Briefly explain your answer.

3.10 **(EOQ with Fixed Batch Sizes)** Suppose that in the EOQ model we can only order batches that are an integer multiple of some number Q_B; that is, we can order a batch of size Q_B, $2Q_B$, $3Q_B$, etc.

a) Prove that, for the optimal order quantity $\hat{Q} = mQ_B$,
$$\sqrt{\frac{m-1}{m}} \le \frac{Q_E}{\hat{Q}} \le \sqrt{\frac{m+1}{m}},$$
where $Q_E = \sqrt{2K\lambda/h}$ is the optimal (non-integer-multiple) EOQ quantity.

b) Suppose that $m \ge 2$ for $\hat{Q}$. Using the result in part (a), prove that $g(\hat{Q}) \le 1.32g(Q_E)$, where $g(\cdot)$ is the EOQ cost function.

c) *Bonus*: Prove that $g(\hat{Q}) \le 1.06g(Q_E)$ (still assuming $m \ge 2$).

3.11 **(Tightness of Power-of-2 Bound)** Prove that the bound given in Theorem 3.3 is tight by developing an instance of the problem such that
$$\frac{f(\hat{T})}{f(T^*)} = \frac{3}{2\sqrt{2}}.$$
Hint: You should be able to come up with a suitable value of T_B in terms of the problem parameters. That is, you should not need to pick values for λ, h, and K; instead, you should be able to leave the values of these parameters unspecified and to express T_B in terms of the parameters to achieve the desired result.

3.12 **(Quantity Discounts for Steel)** Return to Problem 3.1 and suppose that the steel supplier offers the auto manufacturer a price of \$1490 per ton of steel if $Q < 1200$ tons; \$1220 per ton if $1200 \le Q < 2400$, and \$1100 per ton if $Q \ge 2400$. The annual holding cost rate, i, is 0.25.

a) Calculate Q^* and $g(Q^*)$ for the all-units discount structure.

b) Calculate Q^* and $g(Q^*)$ for the incremental discount structure.

3.13 **(Sequence of Q_j^*)** In the EOQ model with incremental quantity discounts, prove that $Q_{j-1}^* < Q_j^*$ for all $j = 1, \ldots, n$.

3.14 **(Sensitivity Analysis for EOQB: Q)** Prove that a result analogous to Theorem 3.2 also describes the sensitivity of the EOQB model with respect to Q; that is, prove that, for any Q:
$$\frac{g(Q,x^*)}{g(Q^*,x^*)} = \frac{1}{2}\left(\frac{Q^*}{Q} + \frac{Q}{Q^*}\right).$$

3.15 **(Sensitivity Analysis for EOQB: x)** In this problem, you will explore the EOQB model's sensitivity to x, the fraction of demand that is backordered.

a) Let $Q(x)$ be the optimal Q for a given x. Derive an expression for $g(Q(x),x)$, the cost that results from choosing an arbitrary value of x and then setting Q optimally.

b) Prove that for any $0 \le x \le 1$,
$$\frac{g(Q(x),x)}{g(Q^*,x^*)} = \sqrt{\frac{(1-x)^2h + x^2p}{x^*p}}.$$

c) Prove that if $h < p$, then for all x,

$$\frac{g(Q(x), x)}{g(Q^*, x^*)} \leq \frac{1}{\sqrt{x^*}}.$$

3.16 **(EOQ with Planned Lost Sales)** Suppose that we are allowed to stock out in the EOQ model, but instead of excess demands being backordered (as in Section 3.5), they are lost. Let x be the fraction of demand that is lost, and let p be the cost per lost sale. Let c be the cost to order each unit. In the standard EOQ and the EOQ with backorders, we could ignore c because each year we order exactly λ items per year on average, regardless of the order quantity Q. But if some demands are lost, we will not order items to replenish those demands; therefore, the total per-unit ordering cost per year *does depend* on the solution we choose.

a) Formulate the total cost per year as a function of Q and x.

b) Prove that

$$x^* = \begin{cases} 0, & \text{if } \lambda(p-c) > \sqrt{2K\lambda h} \\ 1, & \text{if } \lambda(p-c) < \sqrt{2K\lambda h} \\ \text{anything in } [0,1], & \text{if } \lambda(p-c) = \sqrt{2K\lambda h} \end{cases}$$

c) Give an interpretation of the condition $\lambda(p-c) > \sqrt{2K\lambda h}$ and explain in words why the optimal value of x^* follows the rule given in part (b).

d) Part (b) implies that either we meet *every* demand or we stock out on *every* demand—x^* is never strictly between 0 and 1 (except in the special case in which $\lambda(p-c) = \sqrt{2K\lambda h}$). This is not the case in the EOQ with backorders. Explain in words why the two models give different results.

3.17 **(EOQ with Nonlinear Holding Costs)** We assumed that the holding cost for one item in the EOQ model equals ht, where t is the amount of time the item is in inventory. Suppose instead the holding cost for one item is given by he^{bt}, for $b > 0$.

a) Write the average annual cost as a function of Q, $g(Q)$. (Your answer should not include integrals.)

b) Write the first-order condition (i.e., $dg/dQ = 0$) for the function you derived in part (a).

c) The first-order condition cannot be solved explicitly for Q—we can't write an expression like $Q^* =$ [something or other]. Instead, $g(Q)$ must be optimized numerically. Using a nonlinear programming solver, find the Q that minimizes $g(Q)$ using the following parameter values: $\lambda = 500$, $K = 100$, $h = 1$, $b = 0.5$. Report both Q^* and $g(Q^*)$.

Note: As part (e) establishes, $g(Q)$ is quasiconvex everywhere; therefore, you may use a nonlinear solver that relies on this property.

d) Prove that $g(Q)$ is convex at $Q = Q^*$.

Hint: We know the first-order condition says $dg/dQ = 0$ at $Q = Q^*$. Write the second-order condition in such a way that you can make use of the first-order condition.

e) A function f is said to be *unimodal* if there exists some point x^* such that f is increasing on the range $x \leq x^*$ and decreasing on the range $x \geq x^*$. A function

f is said to be *quasiconvex* if $-f$ is unimodal. Prove that $g(Q)$ is quasiconvex for all $Q > 0$.

f) *Bonus*: Prove that $g(Q)$ is convex for all $Q > 0$.

3.18 (EOQ with Batch Demands) Consider an inventory system in which each order is for Q units. Instead of the demand occurring continuously over time (as in the EOQ model), the customer purchases exactly half of the inventory exactly halfway through the order cycle and the remaining half exactly at the end of the order cycle. At that point, a new order is placed, and it arrives instantly. (Therefore, there is no time at which the inventory level equals 0.) The total demand per year is λ, just as in the EOQ model, which means that each order cycle has the same length as in the EOQ model.

a) Write an expression for the average annual total cost.

b) What is the optimal order quantity, Q^*?

3.19 (EOQ vs. EOQB Costs)

a) Prove that the optimal annual holding plus backorder costs in the EOQB model is strictly less than the optimal annual holding cost in the EOQ model.

b) Use part (a) to prove that the total cost (including fixed costs) decreases when we allow backorders.

3.20 (EOQ Generalization) Consider an EOQ-like inventory model whose cost function is given by

$$g(Q) = \frac{aQ^2 + b}{cQ + d} \tag{3.45}$$

for constants $a,\ b,\ c$, and d with $a, c > 0$ and $b, d \geq 0$.

Note that the classical EOQ problem is a special case, since the EOQ cost function (3.3) can be obtained by setting

$$\begin{aligned} a &= h \\ b &= 2K\lambda \\ c &= 2 \\ d &= 0. \end{aligned} \tag{3.46}$$

In this problem you will prove some properties of the cost function (3.45).

a) Prove that

$$Q^* = \frac{\sqrt{a^2d^2 + abc^2} - ad}{ac}.$$

Then show that the classical EOQ model is a special case, i.e., that for the appropriate values of the constants, we get the classical EOQ order quantity.

b) Prove that

$$(Q^*)^2 = \frac{bc - 2adQ^*}{ac}.$$

c) Use part (b) to prove that

$$g(Q^*) = \frac{2a}{c}Q^*.$$

Then show that the classical EOQ model is a special case, i.e., that for the appropriate values of the constants, we have $g(Q^*) = hQ^*$ (Theorem 3.1).

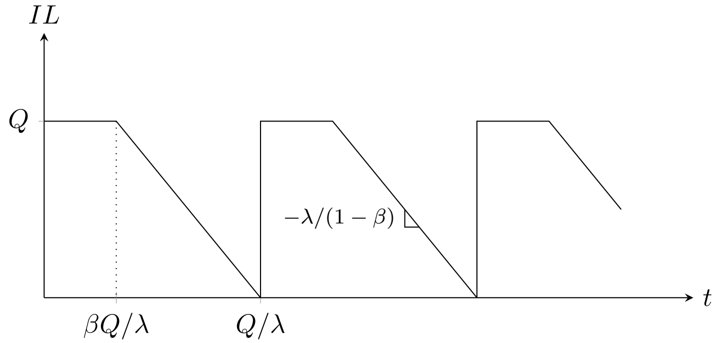


Figure 3.14 Inventory level curve for Problem 3.23.

d) Calculate Q^* and $g(Q^*)$ assuming $a = 20$, $b = 125$, $c = 1.2$, $d = 2.7$.

e) *Bonus*: Prove that

$$\frac{g(Q)}{g(Q^*)} = \frac{1}{2}\left(\frac{Q}{Q^*} + \frac{Q^*}{Q}\right) - [\text{a nonnegative constant}]$$

(analogous to Theorem 3.2), and indicate what the nonnegative constant is. Then show that the classical EOQ model is a special case.

3.21 (Alternate EOQ Proof) Prove that the EOQ cost function can be rewritten as

$$g(Q) = \frac{h}{2\lambda Q}\left(Q - \sqrt{\frac{2K\lambda}{h}}\right)^2 + \sqrt{\frac{2Kh}{\lambda}}.$$

Use this to prove (3.4) *without* using calculus. (Thus, this method provides a proof of the EOQ formula using algebra only.)

3.22 (EPQ for Laundry) A restaurant uses 80 cloth napkins per hour. The napkins are washed by hand at a rate of 110 per hour. Each time the laundry process is started, the restaurant incurs a fixed cost of \$4.00. Napkins in inventory incur a holding cost of \$0.08 per napkin per hour. Stockouts are not allowed. How many napkins should the restaurant have in circulation?

3.23 (EOQ with Zero-Demand Sub-Cycles) Consider the following modification to the EOQ problem. Suppose that, each time an order is placed, the demand is initially 0 for a fraction β of the cycle, and then the demand occurs at a rate of $\lambda/(1-\beta)$ for the duration of the cycle. One can show (you need not) that the total cycle length is still Q/λ, just like in the original EOQ model, and the cycle is divided as shown in Figure 3.14. Calculate the optimal order quantity, Q^*.

3.24 (EOQ with Cycle-Length Costs) Suppose that the inventory ordered in the EOQ problem must be stored in a special piece of storage equipment, and the cost of the equipment depends on the amount of time the inventory will be stored, i.e., the amount of time between replenishment orders. (For example, the product might be perishable; the longer it will be stored in inventory, the more insulation is required in the container.) The storage equipment is leased from a material-handling company. The lease cost per year is

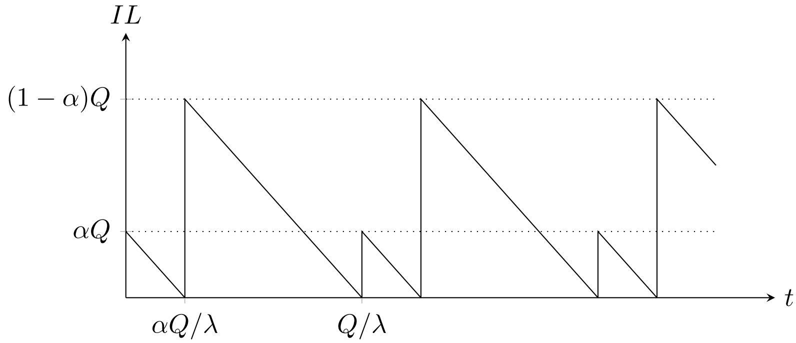


Figure 3.15 Inventory level curve for Problem 3.26.

given by $w \ln T$, where w is a constant, T is the time between consecutive orders, and ln is the natural log function. Holding and fixed costs are still incurred, as in the original EOQ problem. (You can continue to ignore the per-unit purchase cost.)

a) Write the total cost function, $g(Q)$.

b) Write an expression for the optimal order quantity, Q^*.

c) Suppose $h = 2$, $\lambda = 150$, $K = 700$, and $w = 100$. What is Q^*?

d) If $w > 0$, is the optimal order quantity for this model less than, greater than, or equal to that for the original EOQ model?

3.25 (EOQ with Random Half-Orders) Suppose that, in the EOQ model, some orders randomly arrive at only half the requested size. That is, if the order quantity is Q, then the quantity delivered is Q with probability α and $\frac{1}{2}Q$ with probability $1 - \alpha$, for some constant α ($0 \leq \alpha \leq 1$). The remaining parameters and assumptions are as in the standard EOQ model.

a) Determine a closed-form expression for the optimal order quantity, Q^*, as a function of the problem parameters.

b) Will the optimal order quantity in this model be greater than, less than, or equal to that of the classical EOQ? Briefly explain why. (Provide a logical explanation based on the problem, not a mathematical answer based on part (a).)

3.26 (EOQ with Two Deliveries) Consider a variant of the EOQ model in which each order arrives in two separate deliveries. In particular, if we place an order of size Q, then a quantity αQ arrives instantly, and the remaining quantity, $(1 - \alpha)Q$, arrives $\alpha Q/\lambda$ years later, for a fixed constant $0 < \alpha < 1$. Thus, the inventory curve looks like the curve pictured in Figure 3.15.

The fixed cost K is incurred once per order cycle, even though there are two deliveries. As in the standard EOQ, the holding cost is given by h per item per year.

Calculate the optimal order quantity, Q^*.

3.27 (Wagner–Whitin for Aircraft Engines) The Pratt & Whitin Company, which manufactures aircraft engines, needs to decide how many units of a particular bolt to order in order to build engines over the next 4 months. Orders for engines are placed over a year in advance, so the company knows its near-term demand exactly; in particular, the number of engines to produce in the next 4 months will be 150, 100, 80, and 200 in months 1 through 4, respectively. Each engine requires a single bolt. Orders for bolts incur a fixed cost of \$120, and bolts held in inventory incur a holding cost of \$0.80 per bolt per month. Find the optimal order quantities in each period and the optimal total cost.

3.28 (Wagner–Whitin for Sunglasses) The file `sunglasses.xlsx` contains forecast demand (measured in cases) for sunglasses at a major retailer for each of the next 52 weeks. Each order placed to the supplier incurs a fixed cost of \$1100. One case of sunglasses held in inventory for one period incurs a holding cost of \$2.40. Find the optimal order quantities in each period and the optimal total cost.

3.29 (Wagner–Whitin for Glass) A small maker of art glass has orders to make paperweights, vases, and so on over the course of the coming 5 weeks. Based on these orders, it has projected its requirements for its primary raw material—glass rods—over these 5 weeks to be 730, 580, 445, 650, and 880 kg, respectively. Each order to the glass rod supplier incurs a fixed cost of \$100, and each kg of glass rods held in inventory incurs a holding cost of \$0.10 per week.

a) Determine the optimal order quantity in each week, as well as the optimal total cost.

b) Let $\hat{t}$ be the first period in which there is *no* order in your optimal solution from part (a). Suppose the raw material inventory is destroyed at the beginning of period $\hat{t}$ so that the workshop *must* order in period $\hat{t}$. How much should it order in each remaining period of the horizon, and what will be the resulting cost for the entire horizon?

3.30 (Wagner–Whitin Solution from DP #1) Consider the Wagner–Whitin problem with $h = 2$, $K = 50$, $T = 4$, and $(d_1, \ldots, d_4) = (20, 12, 17, 23)$. Suppose you have performed the calculations for $t \geq 2$ and found the following values for θ_t and $s(t)$:

	t			
	2	3	4	5
θ_t	134	96	50	0
$s(t)$	4	5	5	—

Determine which periods to order in, how much to order in each of those periods, and the corresponding optimal cost.

3.31 (Wagner–Whitin Solution from DP #2) Follow the instructions for Problem 3.30 for an instance with $h = 1$, $K = 20$, $T = 4$, and $(d_1, \ldots, d_4) = (25, 15, 15, 30)$, using the following values for θ_t and $s(t)$:

	t			
	2	3	4	5
θ_t	55	40	20	0
$s(t)$	4	4	5	—

3.32 (Wagner–Whitin with Randomly Perishable Goods) Suppose that in the Wagner–Whitin model, all of the items currently held in inventory will perish (be destroyed) with some probability q at the *end* of each time period. For example, if we order 4 periods' worth of demand in period 1, the demand for period 1 will be satisfied for sure, but the inventory consisting of the demand for periods 2–4 will perish with probability q; if it survives (with probability $1 - q$), the inventory for periods 3–4 will perish at the end of period 2 with probability q; and so on. Once the initial ordering schedule is set, no additional orders may be placed.

Obviously, we can no longer require that all demand be satisfied. We will assume that unmet demand is lost (not backordered), and that lost demands incur a penalty cost of p per unit. As in the standard Wagner–Whitin model, we will assume a holding cost of h per unit per time period and a setup cost of K per order.

The sequence of events in each period is as follows:

1. The replenishment order, if any, is placed and is received immediately.
2. Demand occurs and is satisfied from inventory if possible.
3. Remaining inventory either perishes or does not.
4. Holding and stockout costs are incurred based on remaining inventory and lost sales.

a) Show how the arc costs can be computed to capture the new cost function so that the Wagner–Whitin DP algorithm can still be used. Simplify your answer as much as possible.

Hint: The formulas in Section C.5 may come in handy.

b) Illustrate your method by finding the optimal solution for the following 4-period instance: $h = 0.2$, $K = 200$, $p = 3$, $q = 0.25$, and the demands in periods 1–4 are 200, 125, 250, 175. Indicate the optimal solution (order schedule) and the cost of that solution.

c) Do you think the optimal solution to the problem with perishability will involve *more* orders, *fewer* orders, or *the same number* of orders than the optimal solution to the normal Wagner–Whitin problem (without perishability)? Explain your answer.

3.33 (Wagner–Whitin $\rightarrow$ EOQ?) Does the Wagner–Whitin model approach the EOQ model as the length of a time period gets shorter (keeping the total time horizon fixed)? Conduct a small numerical experiment to confirm your answer.

CHAPTER 4

STOCHASTIC INVENTORY MODELS: PERIODIC REVIEW

4.1 INVENTORY POLICIES

In this chapter and the next, we will consider inventory models in which the demand is stochastic. A key concept in these chapters will be that of a *policy*. A policy is a simple rule that provides a solution to the inventory problem. For example, consider a periodic-review model with fixed costs (such as the Wagner–Whitin model) but with stochastic demands. (We will examine such a model more closely in Section 4.4.) One could imagine several possible policies for this system. Here are a few:

1. Every R periods, place an order for Q units.

2. Whenever the inventory position falls to s, order Q units.

3. Whenever the inventory position falls to s, place an order of sufficient size to bring the inventory position to S.

4. Place an order whose size is equal to the first two digits of last night's lottery number. Then, wait a number of periods equal to the last two digits of the lottery number before placing another order.

Now, you probably suspect that some of these policies will perform better (in the sense of keeping costs small) than others. For example, policy 4 is probably a bad one. You

Fundamentals of Supply Chain Theory, Second Edition. Lawrence V. Snyder and Zuo-Jun Max Shen.

Companion website: www.wiley.com/go/Snyder/SupplyChainTheory

probably also suspect that the performance of a policy depends on its *parameters*.[1] For example, policy 1 sounds reasonable, but only if we choose good values for R and Q.

It is often possible (and always desirable) to prove that a certain policy is *optimal* for a given problem—that no other policy (even policies that no one has thought of yet) can outperform the optimal policy, provided that we set the parameters of that policy optimally. For example, policy 3 turns out to be optimal for the model in Section 4.4: If we choose the right s and S, then we are guaranteed to incur the smallest possible expected cost.

When using policies, then, inventory optimization really has two parts: Choosing the form of the optimal policy and choosing the optimal parameters for that policy. Sometimes we can't solve one of these parts optimally, so we use approximate methods. For example, although it's possible to find the optimal s and S for the model in Section 4.4, heuristics are commonly used to find approximately optimal values. Similarly, for some problems, no one even knows the form of the optimal policy, so we simply choose a policy that seems plausible.

We'll consider periodic-review models in this chapter. We'll first consider problems with no fixed costs (in Section 4.3) and then problems with nonzero fixed costs (in Section 4.4). In both of these sections, we'll simply choose a policy to use and focus on optimizing the policy parameters (or, in the case of finite-horizon models, not restrict ourselves to a policy at all). This is the approach taken in the seminal paper by Arrow et al. (1951). Then, in Section 4.5, we'll prove that the policies we chose for the periodic-review models in Sections 4.3 and 4.4 are, in fact, optimal. (We won't prove policy optimality for the continuous-review models in Chapter 5, but those policies, too, are optimal.)

We will continue to use the same notation introduced in Chapter 3. All of the costs we discussed in Section 3.1.3 are in play, including fixed cost K, purchase cost c, holding cost h, and stockout cost p. We'll assume that K and c are nonnegative, that h and p are strictly positive, and that $p > c$ (otherwise it costs more to buy the product from the supplier than it does to stock out, so we should never place an order). Now, however, we'll represent the demand as a random variable D with mean μ, variance σ^2, pdf $f(d)$, and cdf $F(d)$. (D will represent demands over different time intervals in different models, but we'll make this clear in each section.) We'll usually assume that D is a continuous random variable, with a few exceptions.

Throughout most of this chapter, we will assume that unmet demands are backordered. In Section 4.6, we briefly discuss the lost-sales assumption.

Before continuing, we introduce two important concepts in stochastic inventory theory: cycle stock and safety stock. *Cycle stock* (or working inventory) is the inventory that is intended to meet the expected demand. *Safety stock* is extra inventory that's kept on hand to buffer against uncertainty. The target inventory level or order quantity set by most stochastic inventory problems can be decomposed into cycle and safety stock components. We'll see later that the cycle stock depends on the *mean* of the demand distribution, while the safety stock depends on the *standard deviation*.

[1]We don't mean the inputs to the problem, such as costs or demand parameters. Rather, we mean decision variables for the inventory optimization problem, which are often referred to as "parameters."

4.2 DEMAND PROCESSES

In real life, customers tend to arrive at a retailer at random, discrete points in time. Similarly, (some) retailers place orders to wholesalers at random, discrete times, and so on up the supply chain. One way to model these demands is using a *Poisson process*, which describes random arrivals to a system over time. If each customer may demand more than one unit, we might use a *compound Poisson process*, in which arrivals are Poisson and the number of units demanded by each customer is governed by some other probability distribution.

It will often be convenient for us to work with continuous demand distributions (rather than discrete distributions such as Poisson), most commonly the normal distribution with mean μ and variance σ^2. Sometimes, the normal distribution is used as an approximation for the Poisson distribution, in which case $\mu = \sigma^2$ since the Poisson variance equals its mean. (This approximation is especially accurate when the mean is large.)

In the continuous-review case, normally distributed demands mean that the demand over any t time units is normally distributed, with a mean and standard deviation that depend on t. Although it's unusual to think of demands occurring "continuously" in this way, it's a useful way to model demands over time. In the periodic-review case, we simply assume that the demand in each time period is normally distributed.

One drawback to using the normal distribution is that any normal random variable will sometimes have negative realizations, even though the demands that we aim to model are nonnegative. If the demand mean is much greater than its standard deviation, then the probability of negative demands is so small that we can simply ignore it. This suggests that the normal distribution is appropriate as a model for the demand only if $\mu \gg \sigma$—say, if $\mu > 4\sigma$. If this condition fails to hold, then it is more appropriate to use a distribution whose support does not contain negative values, such as the lognormal distribution. (If the true demands are Poisson and we are using the normal distribution to approximate it, then another justification for the condition $\mu \gg \sigma$ is that the normal approximation for the Poisson distribution is most effective when the Poisson mean, λ, is large, in which case $\lambda \gg \sqrt{\lambda}$, which is the standard deviation.)

4.3 PERIODIC REVIEW WITH ZERO FIXED COSTS: BASE-STOCK POLICIES

For the remainder of this chapter, we focus on periodic-review models. The time horizon consists of T time periods; T can be finite or infinite. We will usually assume the lead time is zero, but in Sections 4.3.4.1 and 4.6.2, we'll discuss the implications of assuming a nonzero lead time in the case of backorders (which is easy) and lost sales (which is hard).

We'll first consider the important special case in which $K = 0$ (in this section), and then the more general case of $K \geq 0$ (in Section 4.4). We'll also assume that the costs h, p, c, and K are constant throughout the time horizon.

We will model the time value of money by *discounting* future periods using a discount factor $\gamma \in (0, 1]$. That is, \$1 spent (or received) in period $t + 1$ is equivalent to \$$\gamma$ in period t. If $\gamma = 1$, then there is no discounting. For the single-period and finite-horizon problems, our objective will be to *minimize the total expected discounted cost over the horizon*. However, the total cost over an infinite horizon will be infinite if $\gamma = 1$ and may still be infinite if $\gamma < 1$. Therefore, in the infinite-horizon case, we will minimize the *expected cost per period* if $\gamma = 1$ and the total expected discounted cost over the horizon if $\gamma < 1$. (The solutions to the two problems turn out to be closely related.)

The sequence of events in each period t is as follows:

1. The inventory level is observed.
2. A replenishment order of size Q_t is placed and is received instantly.
3. Demand d_t occurs; as much as possible is satisfied from inventory, and the rest is backordered.
4. Holding and stockout costs are assessed based on the ending inventory level.

The ending inventory level in period t (step 4) is denoted IL_t. It is equal to the starting inventory level in period $t+1$ (step 1) and is given by $IL_t = IL_{t-1} + Q_t - d_t$.

4.3.1 Base-Stock Policies

Throughout Section 4.3, we'll assume that the firm follows a *base-stock policy*.[2] A base-stock policy works as follows: In each time period, we observe the current inventory position and then place an order whose size is sufficient to bring the inventory position up to S. (We sometimes say we "order up to S.") S is a constant—it does not depend on the current state of the system—and is known as the *base-stock level*. Base-stock policies are optimal when $K = 0$; we will prove this in Section 4.5.1. One of the earliest analyses of this type of policy is by Arrow et al. (1951).

In multiple-period models, the base-stock level may be different in different periods. If the base-stock level is the same throughout the horizon, then in every period, we simply order d_{t-1} items. (Why?)

We will divide this problem into three cases—with $T = 1$, $1 \le T < \infty$, and $T = \infty$—and find the optimal base-stock levels in each case.

4.3.2 Single Period: The Newsvendor Problem

4.3.2.1 Problem Statement Consider a firm selling a single product during a single time period. Single-period models are most often applied to *perishable* products, which include (as you might expect) products such as eggs and flowers that may spoil, but also products that lose their value after a certain date, such as newspapers, high-tech devices, and fashion goods. The key element of the model is that the firm only has one opportunity to place an order—before the random demand is observed.

Even if the firm actually sells its products over multiple periods (as is typical), the operations in subsequent periods are not linked: Excess inventory cannot be held over until the next period, nor can excess demands (that is, unmet demands are lost, not backordered). Therefore, the firm's multiperiod model can be reduced to multiple independent copies of the single-period model presented here.

This model is one of the most fundamental stochastic inventory models, and many of the models discussed subsequently in this book use it as a starting point. It is often referred to as the *newsvendor* (or *newsboy*) *model*. The story goes like this: Imagine a newsvendor

[2]Base-stock policies are also sometimes known as *order-up-to policies*, S-*policies*, or $(S-1, S)$-*policies*.

who buys newspapers each day from the publisher for \$0.30 each and sells them for \$1.00. The daily demand for newspapers at his newsstand is normally distributed with a mean of 50 and a standard deviation of 8. If the newsvendor has unsold newspapers left at the end of the day, he cannot sell them the next day, but he can sell them back to the publisher for \$0.12 (called the *salvage value*). The question is: How many newspapers should he buy from the publisher each day? If he buys exactly 50, he has an equal probability of being understocked and overstocked. But it costs more to stock out than to have excess (since stocking out costs him 70 cents in lost profit while excess newspapers cost him $30 - 12 = 18$ cents each). So he should order more than 50 newspapers each day—but how many more?

The inventory carried by the newsvendor can be decomposed into two components: cycle stock and safety stock. As noted in Section 4.1, cycle stock is the inventory that is intended to meet the expected demand—in our example, 50—whereas safety stock is extra inventory that's kept on hand to buffer against demand uncertainty—the amount over 50 ordered by the newsvendor. We will see later that the newsvendor's cycle stock depends on the mean of the demand distribution, while the safety stock depends on the standard deviation.

It is possible for the safety stock to be negative: If stocking out is less expensive than holding extra inventory, the newsvendor would want to order fewer than 50 papers. This can actually occur in practice—for example, for expensive and highly perishable products—but it is the exception rather than the rule.

The mathematical analysis of the newsvendor problem originated with Arrow et al. (1951), though some of the ideas are much older: Edgeworth (1888) uses newsvendor-type logic to determine the amount of cash to keep on hand at a bank to satisfy random withdrawals by patrons. Morse and Kimball (1951) introduced the name "newsboy problem," and Porteus (2008) cites Matt Sobel as proposing the gender-neutral term "newsvendor problem."

As previously noted, the newsvendor model applies to perishable goods. In particular, it applies to goods that perish before the next ordering opportunity. Many perishable goods have a shelf life that exceeds the order interval: For example, a supermarket might place replenishment orders every few days for milk, which has a shelf life of a few weeks. Cases like this are much more difficult to optimize; for a more detailed discussion, see Section 16.3.2.

4.3.2.2 Formulation As usual, we will use h to represent the holding cost: the cost per unit of having too much inventory on hand. In the newsvendor problem, this typically consists of the purchase cost of the unit, minus any salvage value, but may include other costs, such as processing costs. (Since inventory cannot be carried to the next period, this cost is not technically a holding cost, though we will refer to it that way anyway.) Similarly, p represents the stockout cost: the cost per unit of having too little inventory, consisting of lost profit and loss-of-goodwill costs. The holding cost is the cost per unit of positive ending inventory, while the stockout cost is the cost per unit of negative ending inventory. The costs h and p are sometimes referred to as *overage* and *underage* costs, respectively (and some authors denote them c_o and c_u). We can assume that the purchase cost is included in h and that its negative is included in p, and therefore, we assume that $c = 0$. We'll also assume the firm starts the period with $IL = 0$, but this, too, is easy to relax (see Section 4.3.2.6). Since there is only a single period, the discount factor γ won't play a role in the analysis.

We will refer to the model discussed here as the *implicit formulation* of the newsvendor problem since the costs and revenues are not modeled explicitly but instead are accounted for in the holding and stockout costs h and p. (In contrast, see the explicit formulation in Section 4.3.2.4.)

Recall that D is a random variable that represents the demand per period. We'll assume, for now, that D has a continuous distribution. In Section 4.3.2.8, we'll modify the analysis to handle discrete demand distributions.

Our goal is to determine the base-stock level S to minimize the *expected cost* in the single period. The strategy for solving this problem is first to develop an expression for the cost as a function of d (the observed demand) and S (call it $g(S,d)$); then to determine the expected cost $\mathbb{E}_D[g(S,D)]$ (call it $g(S)$); and then (in Section 4.3.2.3) to determine S to minimize $g(S)$.

Let $I(S,d) = (S-d)^+$ and $B(S,d) = (d-S)^+$ be the on-hand inventory and backorders, respectively, at the end of the period if the firm orders up to S and sees a demand of d units. The cost for an observed demand of d is

$$\begin{aligned} g(S,d) &= hI(S,d) + pB(S,d) \\ &= h(S-d)^+ + p(d-S)^+. \end{aligned} \tag{4.1}$$

Since the demand is stochastic, however, we must take an expectation over D. Let $I(S) = \mathbb{E}[I(S,D)]$ and $B(S) = \mathbb{E}[B(S,D)]$ be the *expected* on-hand inventory and backorders if the firm orders up to S. Then,

$$g(S) = hI(S) + pB(S) \tag{4.2}$$

$$\begin{aligned} &= h\mathbb{E}[(S-D)^+] + p\mathbb{E}[(D-S)^+] \\ &= h\int_0^\infty (S-d)^+ f(d)dd + p\int_0^\infty (d-S)^+ f(d)dd \\ &= h\int_0^S (S-d)f(d)dd + p\int_S^\infty (d-S)f(d)dd \end{aligned} \tag{4.3}$$

Let

$$n(x) = \mathbb{E}[(X-x)^+] = \int_x^\infty (y-x)f(y)dy \tag{4.4}$$

$$\bar{n}(x) = \mathbb{E}[(X-x)^-] = \int_0^x (x-y)f(y)dy. \tag{4.5}$$

These functions are known as the *loss function* and the *complementary loss function*,[3] respectively. They can be defined for any probability distribution; here, we define them in terms of the demand distribution. (See Section C.3.1 for more information about these functions.) Then we can rewrite (4.3) as

$$g(S) = h\bar{n}(S) + pn(S). \tag{4.6}$$

[3]The term "complementary loss function" is our own; to the best of our knowledge, this function does not have a name in common usage.

This gives us three ways to write the expected cost function: using $I(S)$ and $B(S)$ (4.2), using integrals (4.3), and using loss functions (4.6). It is also common to use the following identities:

$$I(S) = S - \mu + B(S) \tag{4.7}$$

$$\int_0^S (S-d)f(d)dd = S - \mu + \int_S^\infty (d-S)f(d)dd \tag{4.8}$$

$$\bar{n}(S) = S - \mu + n(S), \tag{4.9}$$

all of which follow from the fact that $x^+ = x + x^-$ for all x. These let us rewrite (4.2), (4.3), and (4.6) as

$$g(S) = h(S-\mu) + (h+p)B(S) \tag{4.10}$$

$$= h(S-\mu) + (h+p)\int_S^\infty (d-S)f(d)dd \tag{4.11}$$

$$= h(S-\mu) + (h+p)n(S). \tag{4.12}$$

4.3.2.3 Optimal Solution The derivatives of the loss function and its complement are given by

$$n'(x) = F(x) - 1 \tag{4.13}$$

$$\bar{n}'(x) = F(x). \tag{4.14}$$

(See Problem 4.23.) Moreover, $n''(x) = \bar{n}''(x) = f(x) > 0$, so $n(\cdot)$ and $\bar{n}(\cdot)$ are both convex, and therefore so is $g(S)$. To minimize $g(S)$, therefore, we set its first derivative to 0. Using (4.6),

$$\frac{dg(S)}{dS} = hF(S) + p(F(S)-1) = (h+p)F(S) - p. \tag{4.15}$$

Setting this equal to 0 gives

$$(h+p)F(S) - p = 0$$

$$\implies F(S) = \frac{p}{h+p} \tag{4.16}$$

$$\implies S^* = F^{-1}\left(\frac{p}{h+p}\right). \tag{4.17}$$

Alternately, we can differentiate (4.12) to get

$$\frac{dg(S)}{dS} = h + (h+p)(F(S)-1) = (h+p)F(S) - p,$$

which is identical to (4.15) and so gives the same optimal solution as (4.17).

The expression for S^* in (4.17) is an important one, so we'll state it as a theorem (which we've now proven).

Theorem 4.1 *The optimal base-stock level for a single-period model with no fixed costs (the newsvendor model) is given by*

$$S^* = F^{-1}\left(\frac{p}{h+p}\right).$$

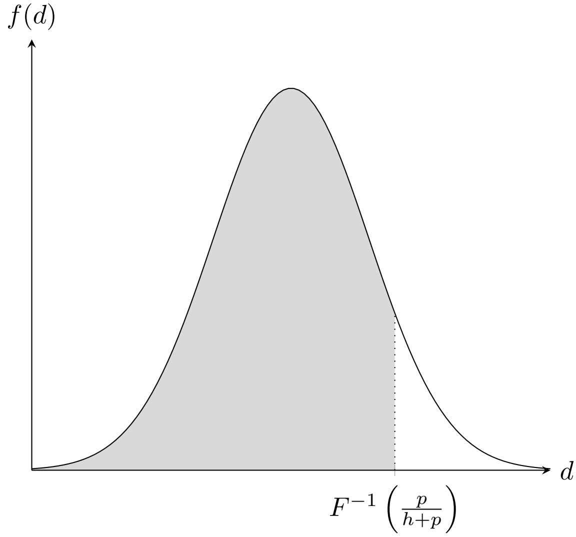


Figure 4.1 Optimal solution to newsvendor problem plotted on demand distribution.

$p/(h+p)$ is known as the *critical ratio* (or *critical fractile*). It is implicit in a result by Arrow et al. (1951) but was first formulated explicitly by Whitin (1953). Since p and h are both positive,

$$0 < \frac{p}{h+p} < 1,$$

so $F^{-1}(p/(h+p))$ always exists. $F(S) = \mathbb{P}(D \leq S)$, or the probability of no stockouts. This is known as the *type-1 service level* (see Section 4.3.4.2). Equation (4.17) then says that under the optimal solution, the type-1 service level should be equal to the critical ratio. It is optimal to stock out in $1-p/(h+p) = h/(h+p)$ fraction of periods. To put it another way, the probability of not having a stockout is equal to the shaded area in Figure 4.1, and Theorem 4.1 says that this area should equal $p/(h+p)$ and that the non-shaded area should equal $h/(h+p)$. As p increases, the critical ratio increases, so S^* and the type-1 service level both increase—it is more costly to stock out, so we should do it less frequently. As h increases, the critical ratio decreases, as does S^*—it is more costly to have excess inventory, so we will order less. The type-1 service level necessarily decreases as well.

Theorem 4.1—or one very much like it—holds for a wide range of models, not just the single-period newsvendor model formulated here. Perhaps most importantly, a variant of the theorem still holds for the mutliperiod, infinite-horizon version of the model; see Section 4.3.4.

☐ **EXAMPLE 4.1**

Cora's Newsstand faces the costs and demand process described in Section 4.3.2.1: a holding cost of $h = 0.18$, a stockout cost of $p = 0.70$, and demand distributed as $N(50, 8^2)$. What is the optimal number of newspapers for Cora to order?

Applying Theorem 4.1, we have

$$S^* = F^{-1}\left(\frac{0.70}{0.18 + 0.70}\right) = F^{-1}(0.795) = 56.6.$$

If Cora can only order an integer number of newspapers, we use (4.6) to calculate $g(S)$ for the neighboring integer values of S. One of the neighboring integer values is guaranteed to be optimal by the convexity of $g(\cdot)$.

$$g(56) = 0.18\bar{n}(56) + 0.70n(56) = 2.0034$$
$$g(57) = 0.18\bar{n}(57) + 0.70n(57) = 2.0000$$

Therefore, the optimal integer number of newspapers is 57.

If the demand distribution is discrete, it is *always* optimal to order the larger neighboring S—see Section 4.3.2.8. But in this example, the demand distribution is continuous even though the order quantity must be discrete, so we must check both $g(S)$ values. □

4.3.2.4 Explicit Formulation The formulation given in Sections 4.3.2.2–4.3.2.3 interprets h and p as the overage and underage costs, respectively—the cost per unit of having too much or too little inventory. The actual cost and revenue parameters are included *implicitly* through the overage and underage costs. For instance, in the example described in Section 4.3.2.1, there is a revenue of \$1.00, a purchase cost of \$0.30, and a salvage value of \$0.12, but these don't appear explicitly in the expected cost function (4.2); rather, they are factored into h and p.

Instead, one can write the expected cost function explicitly using these cost parameters, and the resulting formulation is sometimes more intuitive. In particular, let r be the revenue earned per unit sold, let c be the cost per unit purchased, and let v be the salvage value earned for each unit of excess inventory. We assume $r \geq v$, otherwise we earn more by salvaging a unit than by selling it, so we would never sell any items.

Let h and p be the holding and stockout costs, but reinterpret them to exclude the costs and revenues related to selling, buying, and salvaging the inventory. For example, h might represent a storage cost or a cost to dispose of the inventory; p might represent loss of goodwill or bookkeeping costs related to unmet demands.

As before, the objective is to minimize $g(S)$, which should now include revenues as negative costs. We have:

$$\begin{aligned} g(S) =& cS + h\int_0^S (S-d)f(d)dd + p\int_S^\infty (d-S)f(d)dd \\ & - r\left[(1-F(S))S + \int_0^S df(d)dd\right] - v\int_0^S (S-d)f(d)dd \qquad (4.18) \\ =&(c-r)S + (h+r-v)\int_0^S (S-d)f(d)dd + p\int_S^\infty (d-S)f(d)dd \\ =&(c-r)S + (h+r-v)\bar{n}(S) + pn(S). \qquad (4.19) \end{aligned}$$

Sometimes, this is instead formulated as a *profit maximization* problem in which we maximize $\pi(S) \equiv -g(S)$.

Since $\bar{n}(\cdot)$ and $n(\cdot)$ are both convex, and since $r \geq v$, $g(\cdot)$ is convex (and $\pi(\cdot)$ is concave), so it suffices to set the first derivative of (4.19) to 0:

$$\begin{aligned} \frac{dg}{dS} &= c - r + (h + r - v)F(S) + p(F(S) - 1) = 0 \\ \iff F(S) &= \frac{p + r - c}{h + p + r - v} \\ \iff S^* &= F^{-1}\left(\frac{p + r - c}{h + p + r - v}\right). \end{aligned} \tag{4.20}$$

We can translate this to the implicit version of the problem by determining the overage and underage costs (which we'll denote by h' and p', respectively, since they have a slightly different meaning than h and p in the explicit formulation). For each unit of excess inventory, we incur a holding cost of h, and we paid c for the extra unit but earn v as a salvage value; therefore, $h' = h + c - v$. Similarly, for each stockout, we incur a penalty of p in addition to the lost profit $r - c$, so $p' = p + r - c$. Therefore, applying (4.17), we get

$$S^* = F^{-1}\left(\frac{p'}{h' + p'}\right) = F^{-1}\left(\frac{p + r - c}{h + p + r - v}\right), \tag{4.21}$$

which matches (4.20). The expected cost functions (4.12) and (4.19) are not equal, but they differ only by an additive constant (see Problem 4.15).

It is perfectly acceptable to set any of the cost or revenue parameters to 0 if they are negligible or should not be included in the model.

One word of caution: Avoid mixing the implicit and explicit approaches, since doing so can lead to incorrect accounting of the various costs and revenues. For example, it is a common mistake to use something like the objective function from the implicit formulation (4.3), but to add cS or subtract

$$r\left[(1 - F(S))S + \int_0^S df(d)dd\right]$$

to reflect a purchase cost or sales revenue. If the holding and stockout costs in (4.3) are interpreted as overage and underage costs, then the purchase cost and sales revenue are already implicitly included in h and p (as they are in Example 4.1). To include them explicitly in the objective function would be to double-count them.

□ **EXAMPLE 4.2**

Let us analyze the example in Section 4.3.2.1 using the explicit formulation. We have $r = 1$, $c = 0.3$, and $v = 0.12$. There are no other overage or underage costs (e.g., no disposal costs or loss of goodwill), so $h = p = 0$. Therefore, from (4.20),

$$S^* = F^{-1}\left(\frac{0 + 1 - 0.3}{0 + 0 + 1 - 0.12}\right) = F^{-1}\left(\frac{0.70}{0.88}\right),$$

which is the same optimality condition as in Example 4.1 and yields the same solution: $S^* = 56.6$. □

For the remainder of this chapter and for most of the rest of this book, we will use the implicit formulation. An exception is Chapter 14, which uses something more like the explicit approach.

4.3.2.5 Normally Distributed Demands In this section, we discuss results for the special case in which demands are normally distributed: $D \sim N(\mu, \sigma^2)$, with pdf f and cdf F. We use $\phi(\cdot)$ and $\Phi(\cdot)$ to denote the pdf and cdf, respectively, of the standard normal distribution:

$$\phi(z) = \frac{1}{\sqrt{2\pi}} e^{-z^2/2} \tag{4.22}$$

$$\Phi(z) = \int_{-\infty}^{z} \phi(x)dx \tag{4.23}$$

We also use z_α to denote the αth fractile of the standard normal distribution; that is, $z_\alpha = \Phi^{-1}(\alpha)$.

As discussed in Section 4.2, we will assume that $\mu \gg \sigma$ so that the probability of negative demands is negligible.

From (4.16), we have

$$\begin{aligned} F(S^*) &= \frac{p}{h+p} \\ \iff \Phi\left(\frac{S^* - \mu}{\sigma}\right) &= \frac{p}{h+p} \\ \iff S^* &= \mu + \sigma\Phi^{-1}\left(\frac{p}{h+p}\right). \end{aligned}$$

If we let $\alpha = p/(h+p)$, we have

$$S^* = \mu + z_\alpha \sigma. \tag{4.24}$$

The first term of (4.24) represents the cycle stock—it depends only on μ. The second term represents the safety stock—it depends on σ. The newsvendor problem can be thought of as a problem of setting safety stock. The firm already knows that it will need μ units to satisfy the expected demand; the question is how much more to order to satisfy any demand in excess of the mean. This extra inventory is the safety stock. (See Figure 4.2.)

Note that, as discussed in Section 4.3.2.1, the safety stock is negative if $p < h$ since, in that case, $\alpha < 0.5$ and $z_\alpha < 0$.

We next derive an expression for the expected cost under the optimal solution, as we did with the economic order quantity (EOQ) problem in Section 3.2.3. If X is a normally distributed random variable, then its loss and complementary loss functions are given by

$$n(x) = \mathscr{L}(z)\sigma \tag{4.25}$$

$$\bar{n}(x) = [z + \mathscr{L}(z)]\sigma, \tag{4.26}$$

where $z = (x - \mu)/\sigma$ and

$$\mathscr{L}(z) = \int_{z}^{\infty} (y - z)\phi(y)dy. \tag{4.27}$$

(See Problem 4.22.) (4.25) and (4.26) assume $F(0) = 0$; this is true for actual demands, but it is only approximately true for the normal distribution we use to model demand. $\mathscr{L}(z)$ is called the *standard normal loss function*; it is equivalent to $n(x)$ in (4.4) if X has a standard

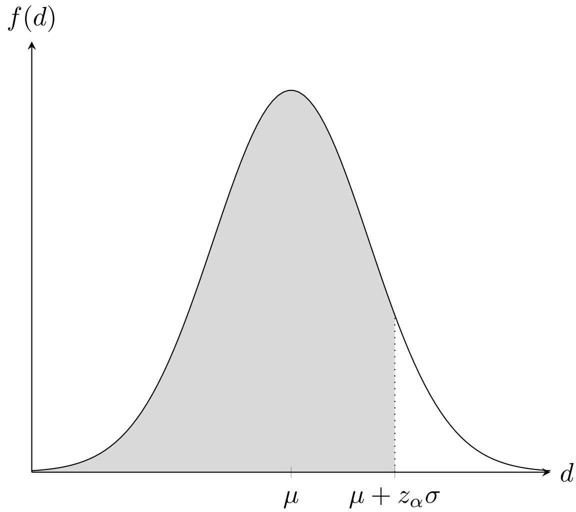


Figure 4.2 Optimal solution to newsvendor problem plotted on normal demand distribution.

normal distribution. $\mathscr{L}(z)$ is tabulated in many textbooks, or it can be computed explicitly as

$$\mathscr{L}(z) = \phi(z) - z\,(1 - \Phi(z))\,. \tag{4.28}$$

Equation (4.28) is convenient for calculating $\mathscr{L}(z)$ in, say, Excel, MATLAB, or Python, since these and many other environments have built-in functions for $\phi(\cdot)$ and $\Phi(\cdot)$ but not for $\mathscr{L}(z)$.

Then, for our problem with normally distributed demands, the cost function (4.6) becomes

$$g(S) = h\,[z + \mathscr{L}(z)]\,\sigma + p\mathscr{L}(z)\sigma = [hz + (h+p)\mathscr{L}(z)]\,\sigma \tag{4.29}$$

for any $S > 0$, where $z = (S - \mu)/\sigma$. From (4.24), $z^* = (S^* - \mu)/\sigma = z_\alpha$. Then from (4.29),

$$\begin{aligned} g(S^*) &= [hz_\alpha + (h+p)\mathscr{L}(z_\alpha)]\,\sigma \\ &= [hz_\alpha + (h+p)\phi(z_\alpha) - (h+p)z_\alpha(1 - \Phi(z_\alpha))]\,\sigma \qquad \text{(from (4.28))} \\ &= [hz_\alpha + (h+p)\phi(z_\alpha) - (h+p)z_\alpha(1-\alpha)]\,\sigma \\ &= [(h+p)\phi(z_\alpha) - (h+p)z_\alpha + (h+p)z_\alpha]\,\sigma \qquad \text{(since } (h+p)\alpha = p\text{)} \\ &= (h+p)\phi(z_\alpha)\sigma \end{aligned} \tag{4.30}$$

It seems surprising at first that (4.30) depends only on σ, not on μ. But with a little reflection, this makes sense: Since the problem comes down to setting safety stock levels, only σ should figure into the objective function. Remember that the objective function includes only holding and stockout costs—costs that result from the randomness in demand, not its magnitude.

Again, let's summarize the optimal order quantity and its cost in a theorem:

Theorem 4.2 *The optimal base-stock level for a single-period model with no fixed costs (the newsvendor model) under demands that are distributed as $N(\mu, \sigma^2)$ is given by*

$$S^* = \mu + z_\alpha\sigma,$$

where $z_\alpha = \Phi^{-1}(\alpha)$ *and* $\alpha = p/(h+p)$. *The optimal cost is given by*

$$g(S^*) = (h+p)\phi(z_\alpha)\sigma.$$

□ **EXAMPLE 4.3**

As in Example 4.1, suppose $D \sim N(50, 8^2)$, $h = 0.18$, and $p = 0.70$. Then $z_\alpha = \Phi^{-1}(0.70/(0.18+0.70)) = 0.8255$. We already know that $S^* = 56.6$ for this problem. We could calculate the optimal cost by plugging S^* into (4.3), or just use (4.30):

$$g(S^*) = (0.18+0.70) \cdot \phi(0.8255) \cdot 8 = 1.9976.$$

□

4.3.2.6 Nonzero Starting Inventory Level We assumed that the firm starts the period with $IL = 0$. In fact, this assumption is easy to relax (and it will be important to do so in the multiperiod versions of this model). If $IL \le S^*$, then the firm should order up to S^*, as usual. But suppose $IL > S^*$. The firm can't order up to S^* since it already has too much inventory. But should the firm order *any* units? By the convexity of $g(S)$, the answer is no: It would be better to leave the inventory level where it is. Therefore, the optimal order quantity at the start of the period is

$$Q = \begin{cases} S^* - IL, & \text{if } IL \le S^* \\ 0, & \text{if } IL > S^*. \end{cases} \tag{4.31}$$

4.3.2.7 Forecasting and Standard Deviations In most real-world settings, we do not know the demand process exactly. Instead, we generate a forecast or estimate of the demand parameters required to make inventory decisions. We'll assume the demand is normally distributed. If we knew μ and σ, we would simply use them in (4.24) to determine the optimal order quantity. But suppose we don't know them; instead, suppose we have observed the demand for a long time, and let d_t be the observed demand in period t. In each period, we can generate an estimate of μ and σ from the historical data. There are many ways to do this (see Chapter 2); one of the simplest is to use a *moving average* (Section 2.2.1) to estimate μ and what we might call a *moving standard deviation* to estimate σ in period t:

$$\hat{\mu}_t = \frac{1}{N}\sum_{i=t-N}^{t-1} d_t \qquad \hat{\sigma}_t = \sqrt{\frac{1}{N-1}\sum_{i=t-N}^{t-1}(d_t - \hat{\mu}_t)^2}$$

To choose an order quantity in period t, we replace μ with $\hat{\mu}_t$ in (4.24). However, it turns out that $\hat{\sigma}_t$ is *not* the right standard deviation to use in place of σ. Instead, the correct quantity to use is the *standard deviation of the forecast error*.

Returning to our historical data, $\hat{\mu}_t$ serves as a forecast for the demand in period t. The *forecast error* (the forecast minus the observed demand in a given period) is a random variable, and it has a mean, denoted μ_e, and a standard deviation, denoted σ_e. The correct quantity to replace σ with in (4.24) is σ_e. We'll omit a rigorous explanation of why this is the case (see, e.g., Nahmias (2005, Section 2.12)), but here is the intuition. The forecasting process introduces sampling error in addition to the randomness in demand, and it is this error that the firm really needs to protect itself against using safety stock. Suppose that the demand is very variable (σ is large), but we are extremely good at predicting it (μ_e and σ_e are both small). We would need very little safety stock, because we can accurately predict how much inventory we will need. Now suppose that the demand is extremely steady (σ is small) but that, for some reason, our forecast is always 100 units too large (μ_e is large, σ_e is small). Here, too, we need very little safety stock, because (knowing our forecast is always too large), we can simply revise our forecast downward. Finally, suppose that the demand is steady (σ is small) but our forecasts are all over the place—sometimes high, sometimes low (μ_e is small, σ_e is large). In this case, we need a lot of safety stock to protect against the uncertainty arising from our inability to predict the demand. In all of these cases, it is the standard deviation of the forecast error that drives the inventory requirement.

Unfortunately, we don't know σ_e any more than we know σ. Instead, we can observe the forecast error in period t,

$$e_t = \hat{\mu}_t - d_t,$$

and estimate the standard deviation of the forecast error as

$$\hat{\sigma}_{e,t} = \sqrt{\frac{1}{N-1} \sum_{i=t-N}^{t-1} (e_t - \hat{\mu}_{e,t})^2},$$

where

$$\hat{\mu}_{e,t} = \frac{1}{N} \sum_{i=t-N}^{t-1} e_t$$

is the estimate of the mean of the forecast error made in period t. (If we know for sure that our forecasts are unbiased, we can replace $\hat{\mu}_{e,t}$ with 0.) We then replace σ with $\hat{\sigma}_{e,t}$ in (4.24) and in the analysis that follows. Of course, if the firm uses a forecasting technique other than moving average, we can simply replace the formulas above with the appropriate ones.

Now, in nearly all of the models in this book (one exception is Section 13.2.2), we assume that the demand parameters are known and stationary. In that case, the forecast $\hat{\mu}_t$ is always equal to the true demand mean μ, and the forecast error is $\mu - d_t$ with mean 0 and standard deviation

$$\hat{\sigma}_{e,t} = \sqrt{\frac{1}{N-1} \sum_{i=t-N}^{t-1} (e_t - \hat{\mu}_{e,t})^2} = \sqrt{\frac{1}{N-1} \sum_{i=t-N}^{t-1} (\mu - d_t)^2},$$

which converges to σ in the long run. Therefore, μ and σ are the appropriate parameters to use.

In general, one can show that $\sigma_e^L = c\sigma^L$ for some constant c (at least for moving average and exponential smoothing forecasts; see, e.g., Hax and Candea (1984, p. 174), or Nahmias (2005, Appendix 2-A)), so in some sense the distinction between the standard deviation of the demand and that of the forecast error is academic, but it's still worth drawing.

This analysis assumes that $\mu_e = 0$, i.e., the forecast is unbiased. If it is not, we should also use $\hat{\mu}_t - \mu_e$ in place of μ in (4.24): If our forecasts tend to be too high ($\mu_e > 0$), then we should reduce the estimate of the mean demand to account for this; and if our forecasts are low ($\mu_e < 0$), we should increase it.

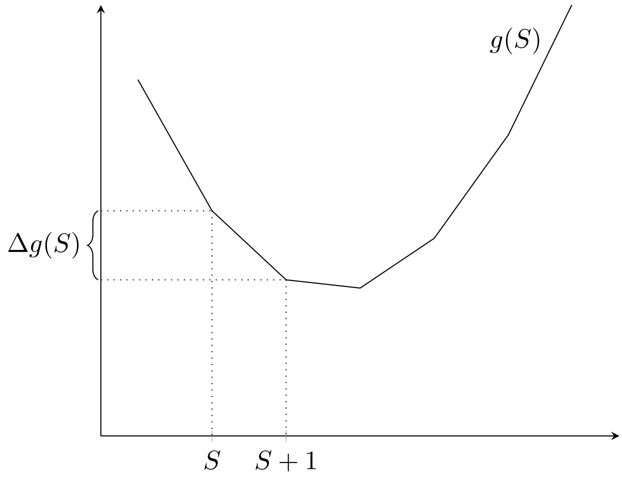


Figure 4.3 $g(S)$ and $\Delta g(S)$.

4.3.2.8 Discrete Demand Distributions Suppose now that D is discrete. In this case, (4.3) becomes

$$g(S) = h \sum_{d=0}^{S} (S-d) f(d) + p \sum_{d=S}^{\infty} (d-S) f(d). \tag{4.32}$$

The expected cost can still be expressed in terms of loss functions, keeping (4.6) as is but replacing the definitions of $n(\cdot)$ and $\bar{n}(\cdot)$ in (4.4) and (4.5) with

$$n(x) = \mathbb{E}[(X-x)^+] = \sum_{y=x}^{\infty} (y-x) f(y) \tag{4.33}$$

$$\bar{n}(x) = \mathbb{E}[(X-x)^-] = \sum_{y=0}^{x} (x-y) f(y). \tag{4.34}$$

(See Section C.3.4 for more on loss functions for discrete distributions.)

The expected cost function $g(S)$ is still convex but no longer differentiable; it is piecewise-linear, with breakpoints at each positive integer. (Why?) Therefore, we cannot use derivatives to minimize it. Instead, we can use *finite differences*. A finite difference is very similar to a derivative except that, instead of measuring the change in the function as the variable changes infinitesimally, it measures the change as the variable changes by one unit. Let

$$\Delta g(S) = g(S+1) - g(S).$$

Imagine starting at $S = 0$ and increasing S one unit at a time. If $g(S+1) < g(S)$, i.e., $\Delta g(S) < 0$, then we would want to increase S to $S+1$ to bring the cost down. Since $g(S)$ is convex, S^* is the smallest S such that $\Delta g(S) \geq 0$. (See Figure 4.3.) Well,

$$\Delta g(S) = h \sum_{d=0}^{S+1} \left((S+1) - d\right) f(d) + p \sum_{d=S+1}^{\infty} \left(d - (S+1)\right) f(d)$$

$$
\begin{aligned}
&- \left[h \sum_{d=0}^{S} (S-d) f(d) + p \sum_{d=S}^{\infty} (d-S) f(d) \right] \\
=& h \sum_{d=0}^{S} f(d) + h \sum_{d=0}^{S} (S-d)\, f(d) - p \sum_{d=S+1}^{\infty} f(d) + p \sum_{d=S+1}^{\infty} (d-S)\, f(d) \\
&- \left[h \sum_{d=0}^{S} (S-d) f(d) + p \sum_{d=S+1}^{\infty} (d-S) f(d) \right] \\
=& hF(S) - p(1-F(S)).
\end{aligned}
$$

Therefore, S^* is the smallest S such that $hF(S) - p(1 - F(S)) \geq 0$; that is:

Theorem 4.3 *The optimal base-stock level for a single-period model with no fixed costs (the newsvendor model) under demands that have a discrete distribution with cdf $F(\cdot)$ is the smallest S such that*

$$
F(S) \geq \frac{p}{h+p}. \tag{4.35}
$$

Unless we get lucky, there is no S such that $F(S)$ *equals* the critical ratio, as it does for continuous demands, so instead we "round up" to the next greater integer. That is, if $F(S-1) < p/(h+p) < F(S)$, there is no need to evaluate both $g(S-1)$ and $g(S)$; $g(S)$ will always be smaller. Note, however, that this does not hold when the demands are *continuous* but the order quantities must be *discrete*; see Problem 4.16.

4.3.3 Finite Horizon

Now consider a multiple-period problem consisting of a finite number of periods, T. Suppose we are at the beginning of period t. Do we need to know the history of the system (e.g., order quantities and demands through period $t-1$) in order to make an optimal inventory decision in period t? The answer is no: All of the information we need to make the inventory decision is contained in a single quantity—the starting inventory level, which equals the ending inventory level in the previous period, IL_{t-1}. Moreover, once we decide how much to order, we can easily calculate the probability distribution of the ending inventory level in period t (as we'll see below). This suggests that the periods can be optimized recursively—in particular, using dynamic programming (DP). Just as in the DP algorithm we used for the Wagner–Whitin problem (Section 3.7.3), this DP will make inventory decisions for period t, assuming that optimal decisions have already been made for periods $t+1, \ldots, T$ and using the cost of those optimal decisions to calculate the cost of the decisions in period t. However, in this DP, the optimal decisions in period t will depend on a random state variable (in particular, IL_{t-1}), whereas the decisions in the Wagner–Whitin DP depended only on the period, t.

First consider what happens at the end of the time horizon. Presumably, on-hand units and backorders must be treated differently now that the horizon has ended than they would be during the horizon. The *terminal cost function*, denoted $\theta_{T+1}(x)$, represents the additional cost incurred at the end of the horizon if we end the horizon with inventory level x. For example, we may incur a *terminal holding cost* h_{T+1} for on-hand units that must be disposed of, and a *terminal stockout cost* p_{T+1} for backorders that must be satisfied through overtime or other expensive measures. Then $\theta_{T+1}(x) = h_{T+1}x^+ + p_{T+1}x^-$. Or, maybe we can salvage excess units at the end of the horizon for a revenue of v_{T+1} per unit, in which case $\theta_{T+1}(x) = -v_{T+1}x^+ + p_{T+1}x^-$.

Let $\theta_t(x)$ be the optimal expected cost in periods $t, t+1, \dots, T$ if we begin period t with an inventory level of x (and act optimally thereafter). We can define $\theta_t(x)$ recursively in terms of $\theta_s(x)$ for later periods s. In each period t, we need to decide how much to order, but we will express this optimization problem not in terms of the order quantity Q, but the *order-up-to level* y, defined as $y = x + Q$.[4] In particular:

$$\theta_t(x) = \min_{y \geq x} \{c(y-x) + g(y) + \gamma \mathbb{E}_D[\theta_{t+1}(y-D)]\}, \tag{4.36}$$

where

$$g(y) = h\int_0^y (y-d)f(d)dd + p\int_y^\infty (d-y)f(d)dd = h\bar{n}(y) + pn(y) \tag{4.37}$$

is the single-period expected cost function (see (4.3) and (4.6)). The minimization considers all possible order-up-to levels $y \geq x$ (since Q must be nonnegative) and, for each, calculates the sum of the cost to order $y - x$ units, the expected cost in period t, and the expected discounted future cost. Note that if we order up to y in period t, then the starting inventory level in period $t+1$ will be $y - D$, where D is the (random) demand in period t; therefore, the (random) cost in periods $t+1, \dots, T$ equals $\theta_{t+1}(y-D)$.

The DP algorithm for the finite-horizon problem is given in Algorithm 4.1. The optimal expected cost for the entire horizon is given by $\theta_1(x_1)$, where x_1 is the inventory level that the system starts with at the beginning of period 1.

Algorithm 4.1 DP for finite-horizon inventory problem

for all x **do** ▷ Calculate terminal costs
    compute $\theta_{T+1}(x)$
end for
for $t = T, \dots, 1$ **do** ▷ Main loop
    for all x **do**
        compute $\theta_t(x)$ using (4.36) ▷ DP recursion
        $y_t(x) \leftarrow$ argmin in right-hand side of (4.36)
    end for
end for
return $\theta_t(x), y_t(x)\ \forall t, x$

One way to think about this DP is as follows. Imagine a spreadsheet whose columns correspond to the periods $1, \dots, T, T+1$ and whose rows correspond to the possible values of x. The value in cell (x, t) equals $\theta_t(x)$. We start by filling in the $\theta_{T+1}(x)$ values in the last column, one for each value of x. Then, we calculate the cells in column T: For each x, we calculate $\theta_T(x)$ using (4.36)—which requires us to look in column $T+1$ for the $\theta_{T+1}(x)$ values—and write the result in cell (x, T). Then we calculate the cells in column $T-1$, using the values in column T, and so on, until we solve period 1. Also imagine a second spreadsheet with identical structure but whose cells contain $y_t(x)$ rather than $\theta_t(x)$.

[4] The order-up-to level y is related to, but not the same as, the base-stock level S. The order-up-to level depends on x: If $x < S$, then $y = S$ and if $x \geq S$ then $y = x$. In contrast, S is a fixed number, independent of x.

The completed spreadsheets tell the firm everything it needs to know about optimally managing the inventory system. If it finds itself with an inventory level of x at the start of period t, it simply looks in the second spreadsheet and orders up to the $y_t(x)$ value that is found in cell (x, t). (The corresponding cell in the first spreadsheet tells the expected current and future cost of this action.)

Two problems with this approach bear mentioning. First, the DP calculates $\theta_t(x)$ "for all x." But x can potentially become arbitrarily large or small, depending on the values we choose for y and on the random demands. For example, if $y_t = 100$ and $D \sim N(100, 10^2)$, it is *possible* (although extremely unlikely) that IL_t will equal $-100{,}000{,}000$, so our spreadsheet should extend at least this far. Of course, this is neither practical nor essential (since the probability is so low), so we typically *truncate* the state space to consider only "reasonable" x values. (The definition of "reasonable" depends on the specific problem at hand.)

Second, even if we consider only a reasonably narrow range of x values, if D has a continuous distribution, there are still an infinite number of possible inventory levels to consider. This problem is typically addressed by *discretizing* the demand distribution so we consider only a finite number of possible demand values. The granularity of the discretization (e.g., do we round demands to the nearest integer? to the nearest 0.001? the nearest 0.000001?) again depends on the specific problem. In general, larger ranges of x values and smaller granularity result in more accurate solutions but longer run times.

Even after we resolve these two problems, this approach is still somewhat unsatisfying, at least from a managerial point of view. The spreadsheets described above will work, but they are fairly cumbersome. It would be nice if we could boil the information contained in the spreadsheets down into a simple policy. To that end, let's look more closely at the results of the DP.

Figure 4.4 plots $y_t(x)$ for three different periods t and for a range of x values for a particular instance of the problem.[5] Essentially, each curve contains the data from a column in the second spreadsheet. Notice that all three curves are flat for a while and then climb linearly along the line $y = x$. That is, for each t, there exists some value S_t such that, for $x < S_t$, we have $y_t(x) = S_t$, and for $x \geq S_t$, we have $y_t(x) = x$. (In particular, $S_1 = 15$, $S_5 = 21$, $S_8 = 17$.) In other words, *these curves each depict a base-stock policy*! In fact, we will prove in Section 4.5.1.2 that a base-stock policy is optimal in every period of the finite-horizon model presented here—the pattern suggested by Figure 4.4 always holds.

Recognizing the optimality of a base-stock policy has simplified the results: We don't need the entire $y_t(x)$ spreadsheet to tell us how to act in each period, we just need a list of S_t values—the optimal base-stock level for each period t. In general, these can be different for different periods, as suggested by Figure 4.4, although in some special cases, the same base-stock level is optimal in every period (see Section 4.5.1.2). However, base-stock optimality has not simplified the computation required to determine the optimal policy—we still need to solve the DP to find the optimal base-stock levels in each period. In particular, S_t is equal to $y_t(-\infty)$, or, assuming we have truncated the range of possible x values, S_t equals $y_t(x)$ for the smallest x value considered.

[5] Actually, for a somewhat more general version of the problem in which the parameters may change (deterministically) over time. The same general results hold for both models.

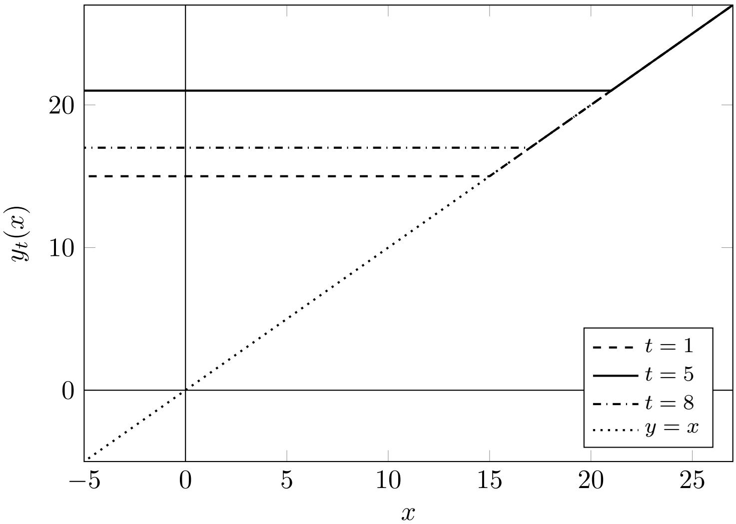


Figure 4.4 DP results, $K = 0$: $y_t(x)$.

4.3.4 Infinite Horizon

Our third and final variety of periodic-review models with no fixed costs is the case in which $T = \infty$. This problem is sometimes referred to as the *infinite-horizon newsvendor model*. If the number of periods is infinite, then the total expected cost across the horizon may be infinite, too. (It certainly will be if $\gamma = 1$.) An alternate objective is to minimize the expected cost per period. The former case is known as the *discounted-cost criterion*, while the latter is known as the *average-cost criterion*. We'll consider the average-cost criterion first, then the discounted-cost criterion.

Under the average-cost criterion, we assume $\gamma = 1$. The expected cost in a given period if we use base-stock level S is given by

$$g(S) = h\int_0^S (S-d)f(d)dd + p\int_S^\infty (d-S)f(d)dd = h\bar{n}(S) + pn(S). \tag{4.38}$$

This is exactly the same expected cost function as in the single-period model of Section 4.3.2. Therefore, the same base-stock level—given in Theorem 4.1—is optimal, in every period!

In formulating (4.38), we glossed over two potentially problematic issues. First, why didn't we account for the purchase cost c, and second, why didn't we account for the cost in future periods? Well, in the long run, the expected number of units ordered is the same—μ—no matter what S we choose. And since $\gamma = 1$, the timing of our orders does not affect the purchase cost. Therefore, the expected purchase cost per period is independent of S.

What about future periods? In (4.38), we didn't account for the impact of our choice of S on future periods. Is this approach sound, or do we need to account for the future cost, as in the finite-horizon DP model of Section 4.3.3? For example, if we start period t with $IL_{t-1} > S_t$, then the expected cost in period t is $g(IL_{t-1})$ rather than $g(S_t)$. In this case, (4.38) would give an incomplete picture of the expected cost in period t since it assumes we can always order up to S. This suggests that we cannot optimize the periods independently.

However, as long as $S_t \le S_{t+1}$, we can be sure that the system starts period $t+1$ with $IL_t \le S_{t+1}$. (Why?) Therefore, no matter what value we choose for S_t, we know that we can always order up to S_{t+1} in period $t+1$. And, as we will see in Section 4.5.1.3, the same base-stock level is optimal in every period. Therefore, $S_t = S_{t+1}$, so $S_t \le S_{t+1}$ and we can optimize (4.38) to find the optimal base-stock level.

Now suppose $\gamma < 1$, i.e., consider the discounted-cost criterion. Since the timing of orders now affects the cost, (4.38) is no longer valid. However, the solution turns out to be nearly as simple: The optimal base-stock level is the same in every period, and it is given by

$$S^* = F^{-1}\left(\frac{p-(1-\gamma)c}{h+p}\right). \tag{4.39}$$

(We omit the proof.)

We summarize these conclusions in the following theorem:

Theorem 4.4 *The optimal base-stock level in every period of an infinite-horizon model with no fixed costs is given by*

$$S^* = F^{-1}\left(\frac{p-(1-\gamma)c}{h+p}\right).$$

Note that this theorem holds for both $\gamma = 1$ and $\gamma < 1$, i.e., for both the average- and discounted-cost criteria.

If demand is normally distributed, then the results from Section 4.3.2.5 still hold, after modifying to account for γ. In particular,

$$S^* = \mu + \sigma\Phi^{-1}\left(\frac{p-(1-\gamma)c}{h+p}\right) = \mu + z_\alpha\sigma, \tag{4.40}$$

where $\alpha = (p-(1-\gamma)c)/(h+p)$. The comments on forecasting in Section 4.3.2.7 also apply here.

4.3.4.1 Lead Times and Reorder Intervals So far, we have assumed that the *lead time* is 0 and that the *reorder interval*—the number of periods that elapse between orders—is 1. (The reorder interval is sometimes called the *review period*.) In this section, we relax those assumptions to allow the lead time to be nonzero and the reorder interval to be greater than 1. In general, we define the *lead-time demand* as the cumulative demand in $L+R$ consecutive periods. In the newsvendor problem in Section 4.3.2, $L=0$ and $R=1$, so the lead-time demand is just the demand in a single period.

The sequence of events is the same as that on page 90. In the discussion that follows, we will use the following notation:

IL_t	= ending inventory level (after step 4 of sequence of events) in period t
IP_t	= inventory position after order is placed but before demand is observed (after step 2 of sequence of events) in period t
D_t	= demand in period t
$D[t,s)$	= cumulative demand in periods $t, t+1, \ldots, s-1$
	$\equiv 0$ if $t > s$
D^τ	= cumulative demand in τ consecutive periods
f^τ, F^τ	= pdf/pmf and cdf of D^τ

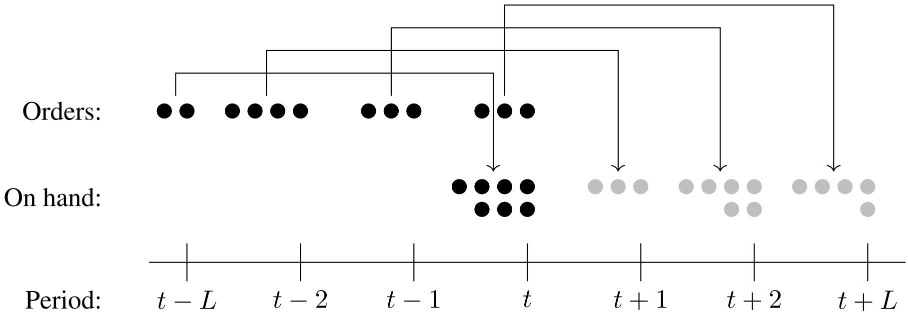


Figure 4.5 Inventory dynamics. All items on order or on hand in period t have arrived by period $t+L$. Items ordered before $t-L$ arrive before t, and items ordered after t arrive after $t+L$. In the figure, $L=3$.

n^τ, $\bar{n}^\tau$ = loss function and complementary loss function of D^τ

For the moment, assume that the reorder interval R equals 1, but allow an L-period lead time, $L \geq 0$. That is, an order placed in period t (in step 2 of the sequence of events) is received in step 2 of period $t+L$. From step 4 of the sequence of events, the holding and stockout costs are incurred based on the ending inventory level, IL, a random variable; therefore, to calculate the expected holding and stockout costs, we need to know the distribution of IL, which in turn depends on the inventory policy parameters (e.g., S). The distribution of IL is not obvious, because IL depends on both the random demand and the inventory actions governed by S. Worse still, there is a delayed reaction: Inventory decisions made in period t do not have an effect on IL until period $t+L$. In the intervening periods, other orders may have arrived (increasing the inventory level) and demands will have occurred (decreasing the inventory level).

The solution to this problem is to relate the inventory level at time $t+L$ to the inventory position at time t (which we know, in the case of a base-stock policy) and to the demand during periods $t, \ldots, t+L$ (whose probability distribution we know). In particular, the ending inventory level in period $t+L$ is given by

$$IL_{t+L} = IP_t - D[t, t+L+1). \tag{4.41}$$

Why is (4.41) true? Well, all of the items included in IP_t—including items on hand and on order—will have arrived by period $t+L$. Moreover, no items ordered after period t will have arrived by period $t+L$. Therefore, all items that are on hand or on order in period t will be included in the ending inventory level in period $t+L$, except for the $D[t, t+L+1)$ items that have since been demanded. (See Figure 4.5.) Another way to think of this is that if the inventory position in period t is IP_t and there is no demand during $[t, t+L]$, then the inventory level in period $t+L$ will be IP_t; and if some demand *does* occur, then IL_{t+L} will be IP_t minus that demand.

Equation (4.41) is a very important equation. It applies to the periodic-review models in this chapter and—in modified form—to the continuous-review models in Chapter 5. The idea dates back to Scarf (1960); Zipkin (2000) refers to it as a *conservation of flow* equation.

Note that (4.41) only holds for the lead time L; that is, in general,

$$IL_{t+L'} \neq IP_t - D[t, t+L'+1). \tag{4.42}$$

This is because some of the units included in IP_t may not be delivered by period $t + L'$ (if $L' < L$), or some units ordered after period t may have been delivered by period $t + L'$ (if $L' > L$).

In steady state, we can drop the time indices from (4.41) and write

$$IL = IP - D^{L+1}, \tag{4.43}$$

where D^{L+1} is a random variable representing the lead-time demand. (We're omitting some of the probabilistic arguments necessary to justify the move from (4.41) to (4.43). See, for example, Galliher et al. (1959) and Zipkin (1986b).) If $L = 0$, then (4.41) simply says that the ending inventory in period t equals the inventory position after the order minus the demand in the period.

Let us apply this insight to the infinite-horizon base-stock problem under the average-cost criterion in Section 4.3.4. (Continue to assume that $\gamma = 1$.) Since this is a base-stock policy, $IP_t = S$ in every period t. Therefore,

$$IL_{t+L} = S - D[t, t + L + 1), \tag{4.44}$$

or in steady state,

$$IL = S - D^{L+1}. \tag{4.45}$$

In other words, the pdf of IL is

$$f_{IL}(x) = f^{L+1}(S - x).$$

The expected cost is still given by (4.38), and Theorem 4.4 still gives the optimal base-stock level, with $f(\cdot)$, $F(\cdot)$, $n(\cdot)$, and $\bar{n}(\cdot)$ replaced by $f^{L+1}(\cdot)$, $F^{L+1}(\cdot)$, $n^{L+1}(\cdot)$, and $\bar{n}^{L+1}(\cdot)$. In essence, we have shifted the accounting so that actions taken in period t do not incur costs until period $t + L$, though all of that logic is buried in the expectations in (4.38).

For normally distributed demands, Theorem 4.4 says that

$$S^* = (L + 1)\mu + \sqrt{L + 1} z_\alpha \sigma, \tag{4.46}$$

where μ and σ refer to the demand per period (and so $(L + 1)\mu$ is the mean and $\sqrt{L + 1}\sigma$ is the standard deviation of lead-time demand). In (4.46), $(L + 1)\mu$ is the cycle stock and $z_\alpha \sqrt{L + 1}\sigma$ is the safety stock. The safety stock is held to protect against fluctuations in *lead time* demand, which is why the safety stock component uses the standard deviation of lead time demand. The reason the cycle stock level depends on the lead time, too, is that the base-stock level refers to the inventory position—so if the lead time is 4 weeks, we always want 4 weeks' worth of cycle stock in the pipeline plus 1 week's worth on hand.

Now let's generalize this logic to allow a reorder interval of $R \geq 1$, so that orders are placed every R periods. Continue to assume that the lead time is $L \geq 0$. The conservation-of-flow argument now goes as follows: Assume that period t is an order period and that $r \in \{0, 1, \ldots, R - 1\}$. All items included in IP_t will have arrived by period $t + L$, and therefore by period $t + L + r$. No items ordered after period t will have arrived by period $t + L + R - 1$ (because any such items would have been ordered in period $t + R$ at the earliest), or therefore by period $t + L + r$. Therefore, the ending inventory level in period $t + L + r$ equals IP_t minus the demand in periods $t, \ldots, t + L + r$:

$$IL_{t+L+r} = IP_t - D[t, t + L + r + 1). \tag{4.47}$$

For a base-stock policy, $IP_t = S$, so we have

$$IL_{t+L+r} = S - D[t, t+L+r+1) \tag{4.48}$$

if t is an order period. Therefore, the expected cost is

$$g(S) = \frac{1}{R} \sum_{r=0}^{R-1} g^{L+r+1}(S), \tag{4.49}$$

where $g(S)$ is the newsvendor cost function (4.3) with $f(\cdot)$ replaced by $f^{L+r+1}(\cdot)$. In general, this cost function must be optimized numerically to find the optimal base-stock level, S^*.

Note that if $R = 1$, then (4.47) and (4.48) reduce to (4.41) and (4.44), respectively, and (4.49) reduces to (4.3).

☐ **EXAMPLE 4.4**

Suppose that Cora's Newsstand also sells city maps, which, coincidentally, incur the same cost and demand structure as in Example 4.1: $h = 0.18$, $p = 0.70$, $D \sim N(50, 8^2)$. The maps are not perishable, so it makes sense for Cora to plan her inventory using an infinite-horizon model. Unmet demands are backordered. Assume that $\gamma = 1$. If $L = 0$, what is the optimal order quantity? What if $L = 4$? What if, in addition, $R = 3$?

If $L = 0$, then the analysis in Example 4.1 remains intact, and we have $S^* = 56.6$. Now suppose $L = 4$. From (4.46),

$$S^* = 5 \cdot 50 + \sqrt{5} \cdot 0.8255 \cdot 8 = 264.8,$$

with cost

$$g(S^*) = (0.18 + 0.70)\phi(0.8255)\sqrt{5} \cdot 8 = 4.47.$$

If, in addition, $R = 3$, then from (4.49) the cost function is

$$g(S) = \frac{1}{3} \sum_{r=0}^{2} g^{5+r}(S).$$

Optimizing numerically, we get $S^* = 344.5$, with cost $g(S^*) = 11.40$. ☐

4.3.4.2 Service Levels The service level measures how successful an inventory policy is at satisfying the demand. There are many definitions of service level. The two most common are as follows:

- *Type-1 service level*: the percentage of order cycles during which no stockout occurs, sometimes called the *cycle service level*, denoted A.

- *Type-2 service level*: the percentage of demand that is met from stock, sometimes called the *fill rate*, denoted B.

(An *order cycle* is the interval between two consecutive orders, or order arrivals. For base-stock policies, the duration of each order cycle is equal to the reorder interval, R. For

Table 4.1 Sample demands and stockouts.

Period	Demand	Stockouts
1	150	0
2	100	0
3	250	50

(s, S) policies, and for the continuous-review models in Chapter 5, the length of an order cycle is stochastic.)

For example, suppose there are 3 periods with the demands and stockouts given in Table 4.1. Then the type-1 service level A is 67% (because we stocked out in 1 of 3 periods), while the type-2 service level B is 90% (because we filled 450 out of 500 demands). In theory, the type-1 service level can be greater than the type-2 service level, but this rarely happens since the type-1 service level is a more stringent measure—any cycle during which a stockout occurs is counted as a "failure," rather than just counting the individual stockouts as failures. (The type-1 service level would be greater than the type-2 service level if, for example, stockouts occur very rarely, but when they do, the number of stockouts is very large.)

Focusing now on base-stock policies, assume that the lead time is $L \geq 0$ periods. If the review period is $R = 1$ (see Section 4.3.4.1), the type-1 service level is easy to calculate: By (4.45), no stockout will occur in a given period if and only if the lead-time demand for the interval ending at that period is less than or equal to S, i.e., $A = F^{L+1}(S)$. If $R > 1$, the type-1 service level is the probability that there are no stockouts in an order *cycle*, i.e., over the R periods between two order arrivals. No stockout occurs in a cycle if and only if the inventory level at the end of the cycle (just before the next order arrival) is positive. By (4.48), this inventory level is positive if and only if $S - D[t, t + L + R) > 0$, which occurs with probability $F^{L+R}(S)$. To summarize:

Theorem 4.5 *The type-1 service level under a periodic-review base-stock policy with lead time $L \geq 0$ and reorder interval $R \geq 1$ is given by*

$$A = F^{L+R}(S),$$

where $F^{L+R}(\cdot)$ is the cdf of the cumulative demand over $L + R$ consecutive periods.

The type-2 service level is a bit trickier. The type-2 service level is

$$B = \mathbb{E}\left[\frac{\text{\# of demands met from stock in a period}}{\text{\# of demands in a period}}\right]. \tag{4.50}$$

We will start by making two simplifying assumptions to derive an approximate expression for the type-2 service level, then relax one and then finally both assumptions to obtain another approximation and an exact expression.

- *Simplifying Assumption 1* (SA1): Backorders never last for more than one order cycle. That is, each arriving order is large enough to clear all existing backorders.
- *Simplifying Assumption 2* (SA2):

$$\mathbb{E}\left[\frac{\text{\# of demands met from stock in a period}}{\text{\# of demands in a period}}\right]$$

$$= \frac{\mathbb{E}[\text{\# of demands met from stock in a period}]}{\mathbb{E}[\text{\# of demands in a period}]}.$$

SA1 is reasonable when S is sufficiently high, as it usually is in practice. SA2 is not true, of course, since $\mathbb{E}[X/Y] \neq \mathbb{E}[X]/\mathbb{E}[Y]$ in general for random variables X and Y; we will explore the loss of accuracy caused by this assumption later in this section. We will use $\hat{B}_1$ to denote the type-2 service level under SA1 and SA2, $\hat{B}_2$ to denote that under SA2 only, and B to denote the exact type-2 service level that assumes neither.

Under SA1 and SA2, we have

$$\begin{aligned}\hat{B}_1 &= \frac{\mathbb{E}[\text{\# of demands met from stock in a period}]}{\mathbb{E}[\text{\# of demands in a period}]} \\ &= \frac{\mathbb{E}[\text{\# of demands met from stock in a cycle}]}{\mathbb{E}[\text{\# of demands in a cycle}]} \\ &= 1 - \frac{\mathbb{E}[\text{\# stockouts in a cycle}]}{\mathbb{E}[\text{\# demands in a cycle}]},\end{aligned} \tag{4.51}$$

where the second equality follows from the fact that each cycle lasts exactly R periods. Assume that an order is placed in period t and consider the cycle that begins in period $t+L$ and ends in period $t+L+R-1$. After the order arrives in period $t+L$, the inventory level is positive (by SA1), so the number of stockouts in the cycle equals $IL^-_{t+L+R-1}$, using the notation in Section 4.3.4.1. Therefore,

$$\begin{aligned}\mathbb{E}[\text{\# stockouts in a cycle}] &= \mathbb{E}[IL^-_{t+L+R-1}] \\ &= \mathbb{E}[(S - D[t, t+L+R))^-] = n^{L+R}(S),\end{aligned}$$

where the second equality follows from (4.48). Therefore,

$$\hat{B}_1 = 1 - \frac{n^{L+R}(S)}{R\mu}. \tag{4.52}$$

Johnson et al. (1995) and subsequent authors refer to this as the "traditional approach."

Now relax SA1. We can no longer calculate the expected number of stockouts in a cycle using the inventory level at the end of the cycle because not all of the "negative" items in $IL_{t+L+R-1}$ are stockouts from the current cycle; some may be left over from the previous cycle. Therefore, we must account for these items more carefully.

Suppose period t is an order period. Let's focus on the cycle that begins in period $t+L$ and ends in period $t+L+R-1$. After the order arrives in $t+L$, no additional orders arrive in this cycle. Therefore, the number of demands met from stock during this cycle equals the difference between the on-hand inventory immediately after the order arrival in period $t+L$ (call this OH_1) and the on-hand inventory at the end of period $t+L+R-1$ (call this OH_2). Moreover, the expected demand during the cycle is $R\mu$. Therefore,

$$\begin{aligned}\hat{B}_2 &= \frac{\mathbb{E}[\text{\# of demands met from stock in a cycle}]}{\mathbb{E}[\text{\# of demands in a cycle}]} \\ &= \frac{\mathbb{E}[OH_1] - \mathbb{E}[OH_2]}{R\mu}.\end{aligned}$$

It remains to evaluate $\mathbb{E}[OH_1]$ and $\mathbb{E}[OH_2]$. First, $OH_1 = X^+$, where X is the inventory level immediately after the order arrival in period $t+L$. Then

$$X = IL_{t+L} + D_{t+L} = S - D[t, t+L) \tag{4.53}$$

since $IL_{t+L} = S - D[t, t+L+1)$ by (4.48). Therefore,

$$\mathbb{E}[OH_1] = \mathbb{E}\left[(S - D[t, t+L))^+\right] = \bar{n}^L(S). \tag{4.54}$$

If $L = 0$, then the right-hand side of (4.53) is S (since $D[t, t+L) \equiv 0$), and in (4.54), $\bar{n}^L(S) = \mathbb{E}[(S - D^L)^+] = S$ since ($D^L = 0$).

Similarly,

$$\mathbb{E}[OH_2] = \mathbb{E}\left[(S - D[t, t+L+R))^+\right] = \bar{n}^{L+R}(S).$$

Therefore,

$$\hat{B}_2 = \frac{\bar{n}^L(S) - \bar{n}^{L+R}(S)}{R\mu}, \tag{4.55}$$

where $\bar{n}^L(S) \equiv S$ if $L = 0$. This approach is due to Hadley and Whitin (1963); see also Johnson et al. (1995), Zhang and Zhang (2007), and Teunter (2009). For another, equivalent, formula for the type-2 service level under SA2, see Problem 4.17.

Since S is chosen to cover $L + R$ periods of demand, we would expect the number of stockouts over L periods to be negligible; put another way,

$$\bar{n}^L(S) = \mathbb{E}\left[\left(D^L - S\right)^-\right] \approx \mathbb{E}\left[-\left(D^L - S\right)\right] = S - \mu L.$$

Therefore, from (4.55),

$$\begin{aligned} B &\approx \frac{S - \mu L - \bar{n}^{L+R}(S)}{R\mu} \\ &= \frac{S - \mu L - \left(S - (L+R)\mu + n^{L+R}(S)\right)}{R\mu} \quad \text{(by (C.14))} \\ &= \frac{R\mu - n^{L+R}(S)}{R\mu} = \hat{B}_1, \end{aligned}$$

which provides another justification of (4.52).

Finally, let us relax both SA1 and SA2 to derive the exact fill rate. As above, assume that t is an order period, and let X be the inventory level after the order arrives at the start of period $t + L$. First assume that $L \geq 1$. Then from (4.53), $X = S - D^L$, i.e., the pdf of X is $f_X(x) = f^L(S - x)$. We will evaluate (4.50) by conditioning on X: By the law of total expectation,

$$\begin{aligned} B =& \mathbb{E}\left[\frac{\text{\# of demands met from stock in a period}}{\text{\# of demands in a period}}\right] \\ =& \mathbb{E}_X\left[\mathbb{E}\left[\frac{\text{\# of demands met from stock in a period}}{\text{\# of demands in a period}} \middle| X\right]\right] \\ =& \int_{x=-\infty}^{S} \left[\int_{d=0}^{\infty} \frac{\min\{x^+, d\}}{d} f^R(d)dd\right] f^L(S-x)dx \\ =& \int_{x=0}^{S} \left[F^R(x) + \int_{d=x}^{\infty} \frac{x}{d} f^R(d)dd\right] f^L(S-x)dx. \end{aligned} \tag{4.56}$$

In the last equality, the change in the lower limit of the first integral comes from the fact that $\min\{x^+, d\} = 0$ for $x < 0$. If $L = 0$, then $X = S$, and (4.56) becomes

$$B = F^R(S) + \int_{d=S}^{\infty} \frac{S}{d} f^R(d)dd. \tag{4.57}$$

If the demands are discrete, the integrals in (4.56) and (4.57) are replaced by sums; see Babiloni et al. (2012).

We summarize the expressions for the type-2 service level in the following theorem.

Theorem 4.6 *For a periodic-review base-stock policy with lead time $L \geq 0$ and reorder interval $R \geq 1$: The approximate type-2 service level under simplifying assumptions SA1 and SA2 (see page 110) is*

$$\hat{B}_1 = 1 - \frac{n^{L+R}(S)}{R\mu}; \tag{4.58}$$

the approximate type-2 service level under simplifying assumption SA2 is

$$\hat{B}_2 = \frac{\bar{n}^L(S) - \bar{n}^{L+R}(S)}{R\mu}, \tag{4.59}$$

where $\bar{n}^L(S) \equiv S$ if $L = 0$; and the exact type-2 service level is

$$B = F^R(S) + \int_{d=S}^{\infty} \frac{S}{d} f^R(d)dd \tag{4.60}$$

if $L = 0$ and

$$B = \int_{x=0}^{S} \left[F^R(x) + \int_{d=x}^{\infty} \frac{x}{d} f^R(d)dd \right] f^L(S-x)dx \tag{4.61}$$

if $L \geq 1$.

Theorem 4.6 holds for both continuous and discrete demands, with the integrals in (4.61) replaced by sums.

☐ **EXAMPLE 4.5**

Suppose that we use a base-stock level of $S = 360$ in the problem with $L = 4$ and $R = 3$ in Example 4.4. What are the type-1 and type-2 service levels?

From Theorem 4.5, the type-1 service level is

$$A = F^7(360) = 0.6817.$$

From Theorem 4.6, the type-2 service level under SA1 and SA2 is

$$\hat{B}_1 \approx 1 - \frac{n^7(360)}{3 \cdot 50} = 0.9709.$$

The type-2 service level under SA2 is

$$\hat{B}_2 = \frac{\bar{n}^4(360) - \bar{n}^7(360)}{3 \cdot 50} = \frac{160.0 - 14.3693}{150} = 0.9709.$$

Since $n^4(360) \approx 10^{-21}$, $\hat{B}_1$ and $\hat{B}_2$ agree to at least 20 decimal places. On the other hand, these both differ a bit from the exact service level, which is

$$B = \int_{x=0}^{360} \left[F^3(x) + \int_{d=x}^{\infty} \frac{x}{d} f^3(d)dd \right] f^4(360-x)dx = 0.9732.$$

☐

Service levels are an important performance measure once the system has been optimized, but they also often play a key role in the optimization itself. The main reason for this is that the stockout penalty p is difficult to estimate, and so it is often preferable to ignore stockouts in the objective function and instead limit them in a constraint, via the service level. That is, we solve a problem of the form

$$\begin{aligned} \text{minimize} \quad & hI(S) && (4.62) \\ \text{subject to} \quad & \text{type-1 service level} \geq \alpha && (4.63) \\ \textit{or}\ \text{subject to} \quad & \text{type-2 service level} \geq \beta. && (4.64) \end{aligned}$$

The objective function comes from (4.2), ignoring the stockout cost. Since $I(S)$ and the service levels are all increasing functions of S, this optimization problem amounts to finding S such that the constraint holds as an equality.

To optimize the base-stock level subject to the type-1 service-level constraint (4.63), we simply have

$$S^* = (F^{L+R})^{-1}(\alpha). \tag{4.65}$$

Since the expressions for the type-2 service level above are more complex than those for type-1, optimizing subject to (4.64) usually requires an iterative search to find the S that satisfies $B = \beta$ in one of the (approximate or exact) expressions for B Theorem 4.6.

☐ **EXAMPLE 4.6**

For the problem setting in Example 4.4 with $L = 4$ and $R = 3$, suppose Cora wishes to require a type-1 service level of 0.9 or a type-2 service level of 0.95. What values of S should she use?

To attain a type-1 service level of 0.9, we use (4.65) to get

$$S^* = (F^7)^{-1}(0.9) = 377.13.$$

For the type-2 service level, let's first use the approximate service level $\hat{B}_1$:

$$\begin{aligned} 1 - \frac{n^7(S)}{5 \cdot 50} &= 0.95 \\ \Longleftrightarrow\ n^7(S) &= 7.5 \\ \Longleftrightarrow\ S &= 351.96. \end{aligned}$$

Now, since

$$\frac{\bar{n}^4(351.96) - \bar{n}^7(351.96)}{3 \cdot 50} = 0.95,$$

this value of S also satisfies the constraint using $\hat{B}_2$. On the other hand, setting $S = 350.83$ makes the exact type-2 service level, B, equal 0.95. ☐

4.4 PERIODIC REVIEW WITH NONZERO FIXED COSTS: (s, S) POLICIES

4.4.1 (s, S) Policies

We now consider the more general case in which the fixed cost K may be nonzero. If $K \neq 0$, it may no longer make sense to order in every period, since each order incurs a cost.

Instead, the firm should order only when the inventory position becomes sufficiently low. In particular, we will assume in this section that the firm follows an (s, S)*-policy*—and in Section 4.5.2, we will prove that such policies are optimal for this system. An (s, S) policy works as follows: In each time period, we observe the current inventory position; if the inventory position is less than or equal to s, then we place an order whose size is sufficient to bring the inventory position up to S. Both s and S are constants, and $s \leq S$. The quantity s is known as the *reorder point* and S as the *order-up-to level*. The reorder point and order-up-to level may change from period to period.

In the special case in which $s = S$, we place an order in every period, and the (s, S) policy is equivalent to a base-stock policy. (In the discrete-demand case, we would use $s = S - 1$; this is why base-stock policies are sometimes known as $(S - 1, S)$ policies.)

Arrow et al. (1951) were the first to formulate the expected cost function for a given choice of the parameters s and S, and to begin the discussion of how to find the optimal s and S. Their analysis simply assumed that the firm followed an (s, S) policy, as we do in this section; the optimality of (s, S) policies for multiperiod problems was not proven until Scarf's (1960) paper.

(s, S) polices are closely related to (r, Q) policies, which we will cover in greater depth in Chapter 5. In an (r, Q) policy, when the inventory position reaches the reorder point, denoted r, we place an order of size Q. The difference is that in an (r, Q) policy, we always order the same quantity (Q), while in an (s, S) policy, we instead order up to a fixed level (S). The two types of policies are equivalent if, in every order cycle, there exists a time at which the inventory position exactly equals the reorder point (s or r), and if we always observe the inventory at that moment. Examples include continuous-review systems with continuously distributed demand (as in Section 5.1) and periodic-review systems in which the demand in each period can only be 0 or 1.

We will discuss how to determine the optimal s and S for the single-period, finite-horizon, and infinite-horizon cases separately, just as we did in Section 4.3 for the zero-fixed-cost case. Actually, the single-period case is not nearly as useful for the $K > 0$ case as it is for the $K = 0$ case. This is because single-period models are most commonly used for perishable products that must be ordered every period; a multiple-period model thus reduces to multiple copies of a single-period one. Even if $K > 0$, we still need to order the perishable product in every period, so the fixed cost becomes a constant and can be ignored. Fixed-cost models are therefore most useful in their multiple-period incarnations. Nevertheless, we will discuss the single-period model to introduce the basic concepts.

4.4.2 Single Period

Suppose the inventory position at the start of the (single) period is x. For given s and S, the ordering rule is: If $x \leq s$, order $S - x$; otherwise, order 0. Once we order (or don't), we incur holding and stockout costs just as in the zero-fixed-cost model, except the base-stock level is replaced by S (if we order) and x (if we don't). Therefore, the total expected cost in the period—as a function of s and S—is given by

$$g(s, S) = \begin{cases} K + g(S), & \text{if } x \leq s \\ g(x), & \text{if } x > s, \end{cases}$$

where $g(S)$ is the expected cost function for the single-period problem with no fixed costs as expressed in (4.3) or (4.6). (As in the single-period model without fixed costs, we are assuming $c = 0$.)

Optimizing $g(s, S)$ over s and S is actually quite easy (Karlin 1958b): We already know from Theorem 4.1 that $F^{-1}(p/(h+p))$ minimizes $g(S)$, so our aim should be to order up to this level unless the fixed cost makes doing so prohibitively expensive. In other words, we should set $S^* = F^{-1}(p/(h+p))$ and set s^* such that $s^* \le S^*$ and $g(s^*) = g(S^*) + K$. (Such an s^* is guaranteed to exist for continuous demand distributions.) Because of the convexity of $g(S)$, if $x \le s$, it is cheaper to order up to S than to leave the inventory position at x, and the reverse is true if $x > s$.

4.4.3 Finite Horizon

The finite-horizon model with nonzero fixed costs can be solved using a straightforward modification of the DP model for the zero-fixed-cost case from Section 4.3.3. Just as before, $\theta_t(x)$ represents the optimal expected cost in periods $t, \ldots, T$ if we begin period t with an inventory level of x (and act optimally thereafter). Now $\theta_t(x)$ must account for the fixed cost in period t (if any), as well as the purchase cost and expected holding and stockout costs in period t, and the expected future costs, as in the $K = 0$ model. In particular,

$$\theta_t(x) = \min_{y \ge x} \{K\delta(y - x) + c(y - x) + g(y) + \gamma \mathbb{E}_D[\theta_{t+1}(y - D)]\}, \tag{4.66}$$

where

$$\delta(z) = \begin{cases} 1, & \text{if } z > 0 \\ 0, & \text{otherwise} \end{cases}$$

and $g(\cdot)$ is as expressed in (4.3) or (4.6).

The DP can be solved exactly as described in Section 4.3.3. Just as in that section, the results of the DP tell us exactly what to order up to in each period t for each starting inventory level x. However, just as before, we would rather have a simple policy to follow, rather than having to specify $y_t(x)$ for every t and x. And, just as before, this is always possible, because a simple policy is always optimal—in this case, an (s, S) policy.

To illustrate this, Figure 4.6 plots $y_t(x)$ for a particular instance of the problem.[6] Just as in Figure 4.4, each curve is flat for a while and then climbs along the line $y = x$. However, whereas in Figure 4.4 the two portions are continuous, here there is a discontinuity representing the point at which we stop ordering. In particular, for period t, there are values S_t and s_t such that for $x \le s_t$, we have $y_t(x) = S_t$, and for $x > s_t$, we have $y_t(x) = x$. In other words, *these curves each depict an* (s, S) *policy*. We will prove in Section 4.5.2.2 that an (s, S) policy is optimal in every period of a finite-horizon model with fixed costs—the pattern suggested by Figure 4.6 always holds.

Once we solve the DP for a given instance, we still need to determine s_t and S_t from the results. This is not difficult: s_t is equal to the largest x such that $y_t(x) = S_t$, and, just as in Section 4.3.3, $S_t = y_t(-\infty)$ (or $y_t(x)$ for the smallest x value considered).

[6]Again, for a variant with time-varying parameters.

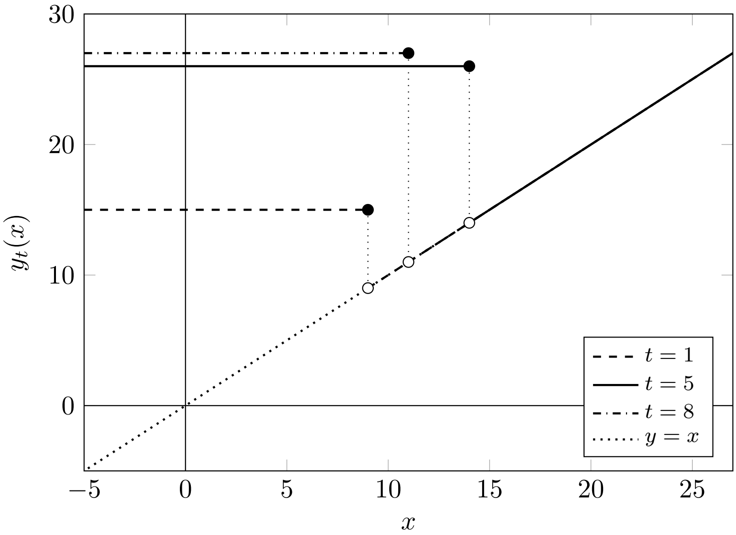


Figure 4.6 DP results, $K > 0$: $y_t(x)$.

4.4.4 Infinite Horizon

Recall that the infinite-horizon model with no fixed costs (Section 4.3.4) is as simple as the single-period model (Section 4.3.2). Unfortunately, this is not true in the fixed-cost case. The infinite-horizon model is more difficult than its single-period or finite-horizon counterparts. To analyze it, we will need a bit of renewal theory.

A *renewal process* is a random variable $N(t)$ that counts the number of "renewals" that have occurred by time t, where the amount of time between the $(n-1)$st renewal and the nth renewals is a random variable X_n. The X_n are independent and identically distributed. (For example, if X_n has an exponential distribution, then the renewals may represent arrivals and N_t is a Poisson arrival process.) Let R_n be a sequence of random variables representing the *reward* that we "earn" at the time of the nth renewal. (R_n may be negative, in which case it is a cost that we pay.) Then

$$R(t) = \sum_{n=1}^{N(t)} R_n$$

is the cumulative reward earned by time t, for $t \geq 0$. We call $R(t)$ a *renewal-reward process*.

The *renewal-reward theorem* gives us an easy way to calculate the long-run expected reward per unit time. Let $\mathbb{E}[X] = \mathbb{E}[X_n]$ and $\mathbb{E}[R] = \mathbb{E}[R_n]$; we will assume that both are finite.

Theorem 4.7 (Renewal-Reward Theorem)

$$\lim_{t\to\infty} \frac{\mathbb{E}[R(t)]}{t} = \frac{\mathbb{E}[R]}{\mathbb{E}[X]} \tag{4.67}$$

Proof. Omitted; see, e.g., Ross (1996). ∎

Returning now to our infinite-horizon inventory model, we may consider a renewal to occur each time an order is placed. Then the time between renewals, X_n, is the length of an order cycle. It has a discrete probability distribution since this is a discrete-time model. The reward at a given renewal is the negative of the cost incurred during the preceding cycle. We are interested in calculating $g(s, S)$, the expected cost per period for given s and S. By the renewal-reward theorem,

$$g(s, S) = \frac{\mathbb{E}[\text{cost per cycle}]}{\mathbb{E}[\text{cycle length}]}, \tag{4.68}$$

where both the numerator and denominator of the right-hand side are functions of (s, S).

Unfortunately, this still leaves us with two problems: (1) The expected cost per cycle and the expected cycle length are not trivial to calculate, and (2) the resulting expected cost function, $g(s, S)$, is not convex. Problem (1) was resolved early on (see, e.g., Veinott and Wagner (1965)), but for decades (2) could not be overcome, and all of the exact algorithms for this problem relied on nearly complete enumeration, with some minor improvements over the years (Veinott and Wagner 1965, Bell 1970, Archibald and Silver 1978). This all changed when Zheng and Federgruen (1991, 1992) introduced a simple, efficient algorithm that finds the exact optimal s and S. It can be viewed as a generalization of the algorithm for (r, Q) policies discussed in Section 5.5.

We'll assume that the per-period demands are drawn iid from a *discrete* (integer) distribution and that the lead time is zero. (Nonzero lead times can be handled using a similar accounting trick as described in Section 4.3.4.1.) We'll further assume that $\gamma = 1$ and consider the average-cost criterion (though Zheng and Federgruen (1991) show how to modify the algorithm for the discounted-cost criterion). We will first derive the cost function $g(s, S)$, then state a few properties of it, and finally describe the algorithm.

Let $M(j)$ be the expected number of periods until the next order is placed, assuming the inventory level[7] equals $s + j$ ($j \geq 1$) after placing the order in step 2 of the sequence of events on page 90. If the inventory level after ordering is $s + j$, then we place an order in the next period if the demand d in the current period is at least j, and otherwise we wait one period and then have a remaining expected wait of $M(j - d)$ periods. Therefore, we can express $M(j)$ recursively as

$$M(j) = \sum_{d=j}^{\infty} f(d) + \sum_{d=0}^{j-1} f(d)(1 + M(j-d)) = 1 + \sum_{d=0}^{j-1} f(d)M(j-d). \tag{4.69}$$

Similarly, let $k(s, y)$ be the total expected cost in the current period through the next order, assuming the inventory level equals $y \geq s$. $k(s, y)$ includes the fixed cost but not the inventory costs in the next order period, and includes the inventory costs but not the fixed cost (if any) in the current period. Using similar logic, we can write $k(s, y)$ recursively as

$$k(s, y) = g(y) + K \sum_{d=y-s}^{\infty} f(d) + \sum_{d=0}^{y-s-1} f(d)k(s, y-d), \tag{4.70}$$

where $g(y)$ is as given in (4.32), since we incur inventory costs of $g(y)$ in the current period and then either place an order in the next period (if $d \geq y - s$) or incur an additional $k(s, y - d)$ in costs (otherwise).

[7] Since the lead time is zero, the inventory level and inventory position are the same.

One can show that the recursive equations (4.69) and (4.70) have the unique solution given by

$$M(0) = 0 \tag{4.71}$$

$$M(j) = M(j-1) + m(j-1) \tag{4.72}$$

$$k(s, y) = K + \sum_{d=0}^{y-s-1} m(d)g(y-d), \tag{4.73}$$

where

$$m(0) = \frac{1}{1 - f(0)} \tag{4.74}$$

$$m(j) = \sum_{d=0}^{j} f(d)m(j-d). \tag{4.75}$$

The expected cost per cycle is $k(s, S)$, and the expected cycle length is $M(S - s)$, so from (4.68),

$$g(s, S) = \frac{K + \sum_{d=0}^{S-s-1} m(d)g(S-d)}{M(S-s)}. \tag{4.76}$$

Let y^* be the minimizer of $g(y)$. We will assume there is only one such minimizer, and only one optimal reorder point and order-up-to level, but the analysis below is easily extended if there are multiple minimizers; see Zheng and Federgruen (1991). The optimal reorder point s^* and order-up-to level S^* lie on either side of y^*:

Lemma 4.8 $s^* < y^* \leq S^*$.

Proof. Omitted; see Veinott and Wagner (1965) and Zheng and Federgruen (1991). ■

The following lemma provides three additional properties of the optimal solution that will be important in the algorithm. First, it gives a condition that lets us identify the optimal reorder point for a given order-up-to level S, denoted $s(S)$. Second, it establishes an efficient way to determine whether one order-up-to level is better than another. Third, it gives an upper bound on S^*.

Lemma 4.9

(a) For a given order-up-to level S, let

$$s = \max\{y < y^* | g(y, S) \leq g(y)\}.$$

Then s is the optimal reorder point for S, i.e., $s = s(S)$.

(b) Let $\hat{S}$ and S be two order-up-to levels. Then $g(s(S), S) < g(s(\hat{S}), \hat{S})$ if and only if $g(s(\hat{S}), S) < g(s(\hat{S}), \hat{S})$.

(c) If (s^*, S^*) are optimal parameters and $g^* = g(s^*, S^*)$ is the corresponding cost, then

$$S^* \leq \max\{y \geq y^* | g(y) \leq g^*\}.$$

Proof. Omitted; see Zheng and Federgruen (1991). ■

Part (a) says that, for fixed S, we can find the optimal reorder point by increasing y until $g(y,S) > g(y)$. Part (b) says that if we have an incumbent order-up-to level $\hat{S}$ and we are considering switching to a new one S, we can tell whether S is better by evaluating S in conjunction with the original reorder point $s(\hat{S})$—we do not have to search for the *best* reorder point for S. Part (c) says that S^* is no larger than the largest y for which $g(y) \le g^*$.

We are now ready to describe Zheng and Federgruen's algorithm. Pseudocode for the algorithm is given in Algorithm 4.2. In the algorithm, S_0 and s_0 are the initial order-up-to level and reorder point, $\hat{s}$ and $\hat{S}$ represent the incumbent solution, and S and s represent a solution under consideration.

Algorithm 4.2 Exact algorithm for periodic-review (s,S) policies with discrete demand distribution (Zheng and Federgruen 1991)

1: $S_0 \leftarrow y^*$ ▷ Set initial S
2: $s \leftarrow y^*$ ▷ Initialize search for $s(S_0)$
3: **repeat** ▷ Search for $s(S_0)$
4:     $s \leftarrow s-1$
5: **until** $g(s,S_0) \le g(s)$
6: $s_0 \leftarrow s$ ▷ Set initial s
7: $\hat{S} \leftarrow S_0$; $\hat{s} \leftarrow s_0$; $\hat{g} \leftarrow g(\hat{s},\hat{S})$ ▷ Initialize incumbent and cost
8: $S \leftarrow \hat{S}+1$ ▷ Choose next order-up-to level to consider
9: **while** $g(S) \le \hat{g}$ **do** ▷ Check for termination via Lemma 4.9(c)
10:     **if** $g(\hat{s},S) < \hat{g}$ **then** ▷ Check for improvement via Lemma 4.9(b)
11:         $\hat{S} \leftarrow S$ ▷ Update incumbent order-up-to level
12:         **while** $g(s,\hat{S}) \le g(s+1)$ **do** ▷ Search for $s(\hat{S})$
13:             $s \leftarrow s+1$
14:         **end while**
15:         $\hat{s} \leftarrow s$; $\hat{g} \leftarrow g(\hat{s},\hat{S})$ ▷ Update incumbent reorder point and cost
16:     **end if**
17:     $S \leftarrow S+1$ ▷ Try next order-up-to level
18: **end while**
19: **return** $(\hat{s},\hat{S})$ ▷ $(\hat{s},\hat{S})$ is optimal

Lines 1–6 identify the initial solution: S_0 is set to y^*, and s_0 is set to the largest $s < S_0$ such that $g(s,S_0) \le g(s)$, which, by Lemma 4.9(a), is optimal for S_0. We set the incumbent solution $\hat{S}$ equal to the initial solution in line 7, and then, in line 8, we choose $S = \hat{S}+1$ as the next order-up-to level to consider.

Next, in lines 9–18, we progressively increment S in search of better order-up-to levels. Line 10 checks whether a given candidate S is better than the incumbent $\hat{S}$; by Lemma 4.9(b), it suffices to compare $g(\hat{s},S)$ to $\hat{g} = g(\hat{s},\hat{S})$. If S improves the cost, we replace the incumbent with it and search for the corresponding optimal s by incrementing s until we have $g(s) \ge g(s,\hat{S}) > g(s+1)$ (lines 12–14), at which point we have found the optimal s for $\hat{S}$ by Lemma 4.9(a). Regardless of whether the new S passed the test in line 10, we move on to the next S (line 17). The **while** loop terminates when $g(S)$ is greater than the incumbent cost, which follows from Lemma 4.9(c): If $g(S) > \hat{g}$, then $g(S) > g^*$, which means S is greater than the maximizer in Lemma 4.9(c) and cannot be optimal. Moreover, all larger S values will also be greater than this maximizer and can be ruled out.

Table 4.2 $g(y)$ for Example 4.7.

y	$g(y)$	y	$g(y)$
0	24.00	8	3.57
1	20.01	9	3.81
2	16.10	10	4.39
3	12.41	11	5.17
4	9.17	12	6.07
5	6.59	13	7.03
6	4.82	14	8.01
7	3.85	15	9.00

☐ **EXAMPLE 4.7**

The daily demand for fruit juice at Cora's Newsstand has a Poisson distribution with mean 6 bottles. Each bottle held in inventory incurs a holding cost of $h = 1$ per day. Unmet demands are backordered and incur a stockout cost of $p = 4$ per bottle per day. To replenish her inventory of fruit juice, Cora must send an employee to pick up the inventory at the supplier, at a labor cost of $K = 5$. Using Algorithm 4.2, find s^* and S^*.

Table 4.2 gives $g(y)$ for $y = 0, 1, \ldots, 15$. From the table, we can see that $y^* = 8$, so we initialize S_0 and s to 8. We have:

$$\begin{aligned} g(7,8) &= 8.56 > g(7) = 3.85 \\ g(6,8) &= 8.49 > g(6) = 4.82 \\ g(5,8) &= 8.33 > g(5) = 6.59 \\ g(4,8) &= 8.20 < g(4) = 9.17 \end{aligned}$$

Therefore, we terminate the **repeat** loop with $s = 4$ and set s_0 to the same. We set $\hat{S} = 8$, $\hat{s} = 4$, and $\hat{g} = g(4, 8) = 8.20$. We set $S = \hat{S} + 1 = 9$ and, since $g(9) = 3.81 < \hat{g}$, we enter the **while** loop at line 9.

We have $g(4, 9) = 8.05 < \hat{g}$, so we update the incumbent $\hat{S}$ to 9 and search for the corresponding optimal s. Since

$$g(4,9) = 8.05 > g(5) = 6.59$$

in line 12, we leave s at 4 and set $\hat{s} = 4$ and $\hat{g} = 8.05$. We then increment S to 10 and return to line 9. Since $g(10) = 4.39 < \hat{g} = 8.05$, we continue the loop. Again the new S is better than the old one since $g(4, 10) = 8.04 < \hat{g}$, so we update $\hat{S} = 10$ and search for the corresponding optimal s. Again, we leave s as it is since

$$g(4,10) = 8.04 > g(5) = 6.59,$$

and we set $\hat{s} = 4$, $\hat{g} = 8.04$, and $S = 11$. In line 9, $g(11) = 5.17 < \hat{g}$, so we continue the loop, but the **if** in the next line fails, because $g(4, 11) = 8.08 > \hat{g}$. The **while** condition holds but the **if** condition fails for $S = 12, 13, 14$, but $g(15) = 9.00 > \hat{g}$, so the loop terminates with $S = 15$. The algorithm terminates with the optimal parameters equal to $(\hat{s}, \hat{S}) = (4, 10)$ and optimal cost 8.04. ☐

There are several heuristics to find near-optimal s and S values. One common approach makes use of the relationship between (s, S) and (r, Q) policies that we discussed in Section 4.4.1: We find the optimal r and Q, either exactly or heuristically—for example, using one of the methods in Section 5.1—and then set

$$
\begin{aligned}
s &= r \\
S &= r + Q.
\end{aligned}
$$

When optimizing the (r, Q) policy, the lead time should be set to $L + 1$ (where L is the lead-time for the (s, S) policy) to account for the difference between continuous and periodic review.

Another approximation involves expressing s and S as explicit functions of the parameters, as follows. Assume that the demand is normally distributed. Let μ and σ^2 be the mean and variance of the single-period demand, and let $\mu_L = \mu L$ and $\sigma_L^2 = \sigma^2 L$ be those of the lead-time demand. Let

$$
Q = 1.30\mu^{0.494} \left(\frac{K}{h}\right)^{0.506} \left(1 + \frac{\sigma_L^2}{\mu^2}\right)^{0.116} \tag{4.77}
$$

$$
z = \sqrt{\frac{Q}{\sigma_L} \frac{h}{p}}. \tag{4.78}
$$

Then set

$$
s = 0.973\mu_L + \sigma_L \left(\frac{0.183}{z} + 1.063 - 2.192z\right) \tag{4.79}
$$

$$
S = s + Q. \tag{4.80}
$$

This approximation is known as the *power approximation* and is due to Ehrhardt and Mosier (1984). It was developed by solving a lot of (s, S) models and fitting regression models for a particular functional form to determine the coefficients. It seems complicated, but it makes some intuitive sense. First, roughly speaking, the parameter Q represents an order quantity. For a moment, suppose $\sigma = 0$ (the demand is deterministic). Then we have

$$
Q = 1.30\mu^{0.494} \left(\frac{K}{h}\right)^{0.506} \approx \sqrt{2}\mu^{0.5} \left(\frac{K}{h}\right)^{0.5} = \sqrt{\frac{2K\mu}{h}},
$$

in other words, the EOQ quantity! Even if $\sigma > 0$, Q is close to the EOQ quantity since the coefficient of the last term in (4.77) has a small exponent. Note also that, since the coefficient in (4.79) is close to 1 and z does not depend on μ, s moves in roughly one-to-one correspondence with μ.

The power approximation performs quite well in practice and has the additional benefit of providing insights into the structure of the optimal solution (such as those in the previous paragraph) that are not obvious when the solution is found using an algorithm. The performance is not as good when $Q/\mu < 1.5$, but a simple modification is available for this case (Ehrhardt 1979).

☐ **EXAMPLE 4.8**

Return to Example 4.4, using $L = 0$, and suppose that $K = 2.5$. Use the (r, Q) approximation and the power approximation to find near-optimal s and S values.

First, the (r, Q) approximation. We can use Algorithm 5.2 to find the optimal parameters; the demand per unit time is $N(50, 8^2)$, and we set the lead time to $L = 1$ to convert to a periodic-review model, so the lead-time demand is also $N(50, 8^2)$. This gives $(r, Q) = (41.29, 45.31)$. Then, we set $s = r = 41.29$ and $S = r + Q = 86.60$.

Alternately, we can use one of the approximate methods to find r and Q. For example, the EOQ+SS approximation (Section 5.3.3) gives $(r, Q) = (56.60, 37.27)$ and $(s, S) = (56.60, 93.87)$.

Now consider the power approximation. We have $\mu = \mu_L = 41.29$ and $\sigma = \sigma_L = 45.31$, so

$$\begin{aligned}
Q &= 1.30\left(50^{0.494}\right)\left(\frac{2.5}{0.18}\right)^{0.506}\left(1 + \frac{8^2}{50^2}\right)^{0.116} = 34.10 \\
z &= \sqrt{\frac{34.10}{8} \cdot \frac{0.18}{0.70}} = 1.0469 \\
s &= 0.973 \cdot 50 + 8\left(\frac{0.183}{1.0469} + 1.063 - 2.192 \cdot 1.0469\right) = 40.20 \\
S &= 40.20 + 34.10 = 74.30.
\end{aligned}$$

We have not discussed an exact algorithm for problems in which the demand has a continuous distribution, as it does in this example. However, we can discretize the demand distribution and then use Algorithm 4.2 to find exact optimal (s, S) values for the discretized problem. Doing so gives $(s, S) = (45, 57)$.

How can we compare the performance of these solutions? We have also not discussed an expected cost function like (4.76) for continuous demand distributions, but again we can discretize the distribution, round the solution, and then apply (4.76) to approximate the cost of a given solution. Doing so on the four solutions above gives the following:

$$\begin{aligned}
&g(41, 87) = 8.08 && ((r, Q) \text{ approximation with exact } (r, Q)) \\
&g(57, 94) = 10.07 && ((r, Q) \text{ approximation with approximate } (r, Q)) \\
&g(40, 74) = 6.80 && (\text{power approximation}) \\
&g(45, 57) = 4.50 && (\text{optimal})
\end{aligned}$$

□

4.5 POLICY OPTIMALITY

Now that we know how to find the optimal S for a base-stock policy (Section 4.3) and the optimal s and S for an (s, S) policy (Section 4.4), we prove that those policy types are in fact optimal for their respective problems. In a way this is a lot to ask—we are trying to show that *no* other policy, of any type, using any parameters, can outperform our chosen policy type (provided we choose the optimal parameters) in the long run. Fortunately, we do not need to prove this explicitly for every possible competing policy type. Rather, we will use the structure of the cost functions to prove that the optimal policy has the desired form.

We will first consider the zero-fixed-cost case, then the fixed-cost case, in both cases considering single-period, finite-horizon, and infinite-horizon cases separately. We will use the same assumptions and notation as in Section 4.4, as well. We continue to assume that the cost and demand parameters are stationary, but the results below still hold if these vary from period to period (deterministically).

Let's focus for a minute on finite-horizon problems with fixed costs. Recall from Section 4.4.3 that $\theta_t(x)$, the optimal cost in periods $t, \ldots, T$ if we begin period t with an inventory level of x, can be calculated recursively as

$$\theta_t(x) = \min_{y \geq x} \{K\delta(y-x) + c(y-x) + g(y) + \gamma \mathbb{E}_D[\theta_{t+1}(y-D)]\}, \tag{4.81}$$

where $g(y)$ is given by (4.3) or (4.6). The zero-fixed-cost problem is a special case, obtained by setting $K = 0$, and the single-period problem is also a special case, obtained by setting $T = 1$. Note that (4.81) does not assume that any particular policy is being followed. It simply determines the optimal action (order-up-to level) for each starting inventory level x in each period t. Our goal throughout this section will be to use the structure of (4.81) to show that the optimal actions follow the policies we have conjectured are optimal.

4.5.1 Zero Fixed Costs: Base-Stock Policies

We first consider the case in which $K = 0$ and prove that—regardless of the horizon length—a base-stock policy is always optimal. These results date back to Karlin (1958a, 1960) and Veinott (1965), among others.

4.5.1.1 Single Period In this section, we'll consider the special case in which $T = 1$ and $K = 0$. We'll also assume that the terminal cost function (see Section 4.3.3) is equal to 0. This assumption is not necessary—we could instead assume only that the terminal cost function is convex—but it simplifies the analysis.

Under these assumptions, (4.81) reduces to

$$\theta(x) = \min_{y \geq x} \{c(y-x) + g(y)\}. \tag{4.82}$$

Of course, we already know how to solve this problem: Theorem 4.1 gives the optimal solution. But our goal here is not to find the optimal solution for a given instance, but rather to prove that the optimal solution always has a certain structure—a base-stock policy.

It will be useful to keep the parts of (4.82) that depend on x separate from those that don't. To that end, we can rewrite $\theta(x)$ as

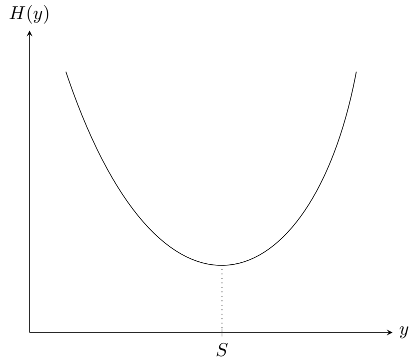
$$\theta(x) = \min_{y \geq x} \{H(y) - cx\}, \tag{4.83}$$

where

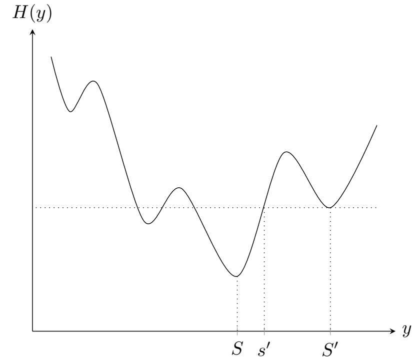
$$H(y) = cy + g(y). \tag{4.84}$$

Since we are calculating $\theta(x)$ for fixed x, from (4.83), we see that the optimal decision can be found by minimizing $H(y)$ over $y \geq x$—that is, starting at $y = x$, we want to minimize $H(y)$ looking only "to the right" of x. The question is, does this strategy give rise to a base-stock policy?

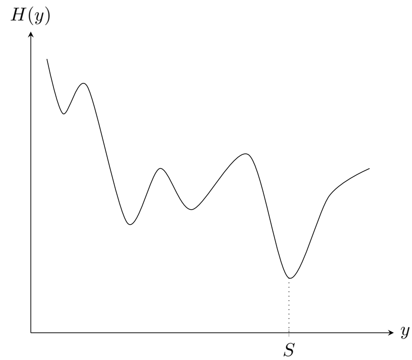
Suppose $H(y)$ has a shape similar to that pictured in Figure 4.7(a). In this example, $H(y)$ is minimized at $y = S$. If $x < S$, then the optimal strategy is to set $y = S$, while if $x \geq S$, the optimal strategy is to do nothing—to set $y = x$. In other words, the optimal policy is a base-stock policy. This argument works for any convex function $H(y)$—if $H(y)$ is convex, then a base-stock policy is optimal. And $H(y)$ is convex because $g(y)$ is convex, so we have now sketched the proof of the following theorem.



(a) $H(y)$ convex; base-stock policy is optimal.



(b) $H(y)$ nonconvex; base-stock policy is not optimal.



(c) $H(y)$ nonconvex; base-stock policy is still optimal.

Figure 4.7 Hypothetical shapes of the function $H(y)$.

Theorem 4.10 *A base-stock policy is optimal for the single-period problem with no fixed costs.*

What if $H(y)$ is nonconvex? (This would happen if we chose some other single-period expected cost function $g(y)$.) For example, suppose $H(y)$ has a shape similar to that in Figure 4.7(b). Then a base-stock policy is *not* optimal since for $x < S$, we would set $y = S$, while for $x \in (s', S']$, we would set $y = S'$. On the other hand, there are nonconvex functions for which a base-stock policy is still optimal—the function in Figure 4.7(c) is an example. Even though the function has several local minima, it is still optimal to order up to S if $x < S$ and to do nothing otherwise.

4.5.1.2 Finite Horizon It was simple to prove that $H(y)$ is convex, and therefore that a base-stock policy is optimal, for the single-period problem. Our main goal in this section will be to prove that the analogous functions (one per period) are also convex. This is a bit trickier than in the single-period case.

The finite-horizon, zero-fixed-cost version of (4.81) is

$$\theta_t(x) = \min_{y \geq x} \{c(y-x) + g(y) + \gamma \mathbb{E}_D[\theta_{t+1}(y-D)]\}. \tag{4.85}$$

Here, we allow the terminal cost function $\theta_{T+1}(\cdot)$ to be nonzero, and we'll add the requirement that it is convex.

Again we rewrite $\theta_t(x)$ to separate the parts that depend on x from those that don't:

$$\theta_t(x) = \min_{y \geq x} \{H_t(y) - cx\}, \tag{4.86}$$

where

$$H_t(y) = cy + g(y) + \gamma \mathbb{E}_D[\theta_{t+1}(y-D)]. \tag{4.87}$$

It is simple to argue that, if $H_t(y)$ is convex, then a base-stock policy is optimal in period t. The tricky part is showing that $H_t(y)$ is convex for every t. We'll prove this recursively in the next lemma, showing that if $\theta_{t+1}(x)$ is convex, then so are $H_t(y)$ and $\theta_t(x)$. Then, in Theorem 4.12, we'll get the recursion started, implying that all the $H_t(y)$ functions are convex and that a base-stock policy is optimal in every period.

Lemma 4.11 *If $\theta_{t+1}(x)$ is convex, then:*

(a) $H_t(y)$ is convex.

(b) A base-stock policy is optimal in period t, and any minimizer of $H_t(y)$ is an optimal base-stock level.

(c) $\theta_t(x)$ is convex.

Proof.

(a) Clearly cy is convex since it is linear, and we know from Section 4.3.2.3 that $g(y)$ is convex. The third term is convex because $\theta_{t+1}(x)$ is convex (by assumption) and expectation preserves convexity.[8] Therefore, $H_t(y)$ is convex, since the sum of convex functions is convex.

(b) From (a), we know that $H_t(y)$ is convex. Let S_t be a minimizer of $H_t(y)$. If $x < S_t$, then the optimal $y \geq x$ is at $y = S_t$; if $x \geq S_t$, then H_t is nondecreasing to the right of x (by convexity), so the optimal $y \geq x$ is $y = x$. This is exactly the definition of a base-stock policy.

(c) From (4.86), $\theta_t(x)$ is the minimum over y of $H_t(y)$ (minus a constant). Since minimization preserves convexity,[9] the convexity of $H_t(y)$ from (a) implies that of $\theta_t(x)$. ∎

[8]This is a well-known property of convex functions. It says that, if $f(x)$ is a convex function and Y is a random variable, then $\mathbb{E}_Y[f(x-Y)]$ is convex.

[9]Another well-known property of convex functions: If $f(x,y)$ is convex and $g(x) = \min_y\{f(x,y)\}$, then $g(x)$ is convex (Boyd and Vandenberghe 2009, Section 3.2.5).

We have done most of the heavy lifting, but we're not done yet. All we have shown is that a base-stock policy is optimal in period t *if* $\theta_{t+1}(x)$ is convex. The next theorem establishes our main result—that a base-stock policy is optimal in every period—and the convexity of $\theta_{T+1}(\cdot)$ gets the recursion started.

Theorem 4.12 *If the terminal cost function $\theta_{T+1}(x)$ is convex, then a base-stock policy is optimal in each period of the finite-horizon problem with no fixed costs.*

Proof. By assumption, $\theta_{T+1}(x)$ is convex. Therefore, by Lemma 4.11(b), a base-stock policy is optimal in period T. Moreover, $\theta_T(x)$ is convex by Lemma 4.11(c). This implies that a base-stock policy is optimal in period $T-1$ and that $\theta_{T-1}(x)$ is convex. Continuing this logic, a base-stock policy is optimal in every period. ■

This proof assumed that the single-period cost function, $g(y)$, is convex. In fact, it is sufficient to assume the slightly weaker condition that $g(y)$ is *quasiconvex*, i.e., that $-g(y)$ is *unimodal*—in other words, that $g(y)$ has a unique local (and therefore global) minimum. For a proof, see Veinott (1966).

Of course, this analysis says nothing about how to find the optimal base-stock levels. In general, we need to use the DP from Section 4.3.3 to find those. In most cases, the base-stock levels will change over time, and the pattern depends on what happens at the end of the horizon, i.e., the terminal cost function. For example, suppose backorders that are outstanding at the end of the horizon must be cleared by, say, air-freighting inventory from overseas at a very high cost. Then the base-stock levels will increase at the end of the horizon to prevent these costly backorders. Conversely, suppose the product in question is a hazardous material that must be disposed of at a very high cost if any remains at the end of the horizon. Then the base-stock levels will decrease at the end of the horizon to ensure that the inventory is sold. But if the terminal cost function is just right, the same base-stock level will be optimal in every period. Moreover, in this special case, the optimal base-stock levels can be found explicitly, without requiring an algorithm. This policy is called a *myopic policy* because it optimizes only a single period at a time, ignoring the rest of the horizon. In this special case, then, the myopic policy is optimal in every period.

The special case is defined by setting the terminal cost function to

$$\theta_{T+1}(x) = -cx.$$

This terminal cost function would be applicable if, for instance, at the end of the horizon, any excess inventory can be returned to the supplier for a full reimbursement of the order cost c and any backorders must be cleared by purchasing a new item, again at a cost of c.

First consider period T, for which it is straightforward to find the optimal base-stock level:

$$\begin{aligned} H_T(y) &= cy + g(y) + \gamma \mathbb{E}_D[\theta_{T+1}(y-D)] \\ &= cy + g(y) + \gamma \mathbb{E}_D[-c(y-D)] \\ &= c(1-\gamma)y + g(y) + \gamma c\mu, \end{aligned}$$

where $\mu = \mathbb{E}[D]$. The optimal base-stock level is a minimizer of $H_T(y)$, so we set $H_T'(y) = 0$:

$$H_T'(y) = c(1-\gamma) + (h+p)F(y) - p = 0$$

(from (4.15)), or

$$F(y) = \frac{p - (1-\gamma)c}{h+p}.$$

The optimal base-stock level in period T is therefore

$$S_T^* = F^{-1}\left(\frac{p - (1-\gamma)c}{h+p}\right). \tag{4.88}$$

This is the same solution as the infinite-horizon newsvendor model in Theorem 4.4.

Now we know that (4.88) gives the optimal base-stock level in period T; it remains to show that the same base-stock level is optimal in the other periods. In period T, the solution to the minimization in (4.86) is to set $y = S_T^*$ if $x \le S_T^*$ and $y = x$ otherwise. Therefore,

$$\theta_T(x) = \begin{cases} H_T(S_T^*) - cx, & \text{if } x \le S_T^* \\ H_T(x) - cx, & \text{otherwise.} \end{cases} \tag{4.89}$$

Now let's compute $H_{T-1}(y)$ in order to derive the optimal base-stock level for period $T-1$. From (4.87),

$$\begin{aligned} H_{T-1}(y) &= cy + g(y) + \gamma \mathbb{E}_D[\theta_T(y-D)] \\ &= \begin{cases} cy + g(y) + \gamma \mathbb{E}_D[H_T(S_T^*) - c(y-D)], & \text{if } y \le S_T^* \\ [\text{something else}], & \text{if } y > S_T^*. \end{cases} \end{aligned} \tag{4.90}$$

The first case holds because if $y \le S_T^*$, then surely $y - D \le S_T^*$, and therefore, the first case in (4.89) holds. But the second case is harder because if $y > S_T^*$, then the first case in (4.89) will hold for some D, and the second case will hold for others. Fortunately, it will turn out that we won't need to write out an expression for the second case of (4.90): If we can show that the derivative of $H_{T-1}(y)$ is 0 for some $y \le S_T^*$, then by the convexity of $H_{T-1}(y)$ (Lemma 4.11(a)), that y minimizes $H_{T-1}(y)$ and we can ignore the case in which $y > S_T^*$. So assume that $y \le S_T^*$. Then

$$\begin{aligned} H_{T-1}(y) &= cy + g(y) + \gamma \mathbb{E}_D[H_T(S_T^*) - c(y-D)] \\ &= c(1-\gamma)y + g(y) + \gamma c\mu + \gamma H_T(S_T^*), \end{aligned}$$

which differs from $H_T(y)$ only by an additive constant. Therefore, its derivative equals 0 for the same value of y, and we have the same optimal base-stock level. Continuing this logic backwards, we get the following theorem:

Theorem 4.13 *If $\theta_{T+1}(x) = -cx$, then the myopic base-stock level, given by*

$$S^* = F^{-1}\left(\frac{p - (1-\gamma)c}{h+p}\right),$$

is optimal in every period.

The optimal base-stock level in Theorem 4.13 is identical to the infinite-horizon base-stock level from Theorem 4.4.

4.5.1.3 *Infinite Horizon* Now suppose that $T = \infty$. The main result is the following:

Theorem 4.14 *A base-stock policy is optimal in each period of the infinite-horizon problem with no fixed costs.*

And we already know the optimal base-stock level, from Theorem 4.4. We will omit the proof of Theorem 4.14. It uses many of the ideas from the earlier proofs and is not very difficult (see, e.g., Zipkin 2000).

4.5.2 Nonzero Fixed Costs: (s, S) Policies

We now allow $K \neq 0$ and prove that an (s, S) policy is optimal. We will present formal proofs for the single-period and finite-horizon cases but only state the result without proof for the infinite-horizon case. In the single-period case, we will argue that an (s, S) policy is optimal using the convexity of $H(y)$, just as we used the convexity of this function to prove that a base-stock policy is optimal for the zero-fixed-cost case. However, in the finite-horizon problem, $H_t(y)$ is no longer convex (except for $t = T$). Fortunately, however, it is close enough to convex (in a specific way to be made more precise later) to establish the result.

4.5.2.1 *Single Period* Assume that $T = 1$ and (as in Section 4.5.1.1) that the terminal cost function equals 0. Then (4.81) reduces to

$$\theta(x) = \min_{y \geq x}\{K\delta(y - x) + c(y - x) + g(y)\} \tag{4.91}$$

$$= \min_{y \geq x}\{K\delta(y - x) + H(y) - cx\}, \tag{4.92}$$

where $H(y)$ is the same as before, as defined in (4.84).

Let S^* be the minimizer of $H(y)$. Since $H(y)$ is convex, we should definitely not order if $x > S^*$. What if $x \leq S^*$? We may not even wish to order in this case—it depends on how much we save by ordering versus how much it costs to order. That is, we should order up to S^* if

$$H(x) - H(S^*) \geq K \tag{4.93}$$

and do nothing otherwise. Which values of x satisfy (4.93)? By the convexity of $H(y)$, there exists an s^* such that all $x \leq s^*$ satisfy (4.93). In particular, s^* is the x such that $H(x) - H(S^*) = K$. (There may be multiple such x if $H(y)$ is not strictly convex. However, if the demand cdf $F(\cdot)$ is strictly increasing, then $g(y)$ and hence $H(y)$ are strictly convex.)

We have now proved the following result, initially due to Karlin (1958b):

Theorem 4.15 *An (s, S) policy is optimal for the single-period problem with fixed costs.*

And, as we argued in Section 4.4.2, S^* is the minimizer of $H(y)$ and $s^* \leq S^*$ satisfies $H(s^*) - H(S^*) = K$.

4.5.2.2 *Finite Horizon* Recall the logic for proving that a base-stock policy is optimal for the finite-horizon model with no fixed costs (Lemma 4.11 and Theorem 4.12): Since $\theta_{T+1}(x)$ is convex, so is $H_T(y)$; therefore, a base-stock policy is optimal in period T and $\theta_T(x)$ is convex; therefore, $H_{T-1}(y)$ is convex; therefore, a base-stock policy is optimal in

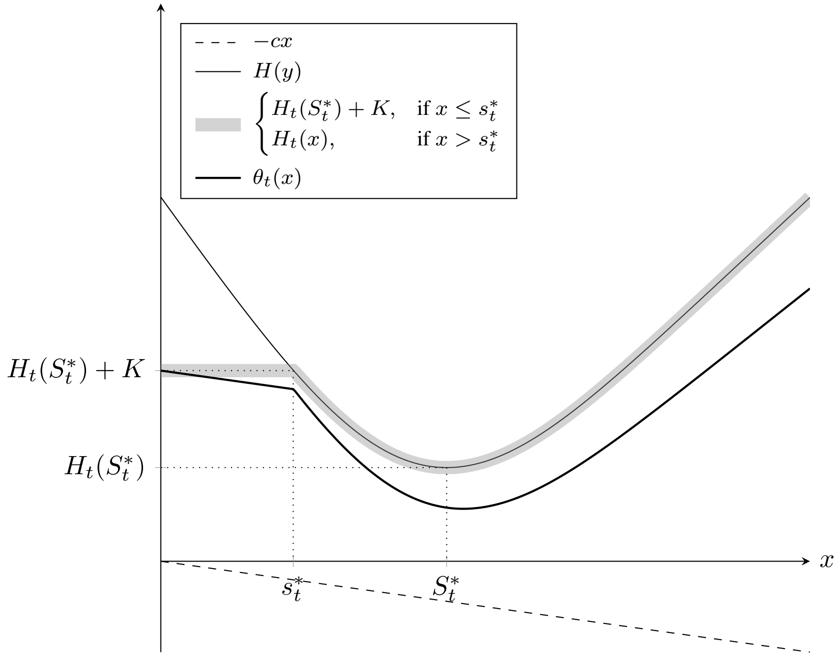


Figure 4.8 Nonconvexity of $\theta_t(x)$.

period $T-1$, and $\theta_{T-1}(x)$ is convex; and so on. Unfortunately, the convexity implications break down when fixed costs are present. Let's see why.

From (4.81),

$$\begin{aligned} \theta_t(x) &= \min_{y \ge x} \{K\delta(y-x) + c(y-x) + g(y) + \gamma \mathbb{E}_D[\theta_{t+1}(y-D)]\} \\ &= \min_{y \ge x} \{H_t(y) + K\delta(y-x) - cx\}, \end{aligned}$$

where $H_t(y)$ is as defined in (4.87). Let's assume that $H_t(y)$ is convex. Is $\theta_t(x)$? Since $H_t(y)$ is convex, an (s, S) policy is optimal in period t. This implies that

$$\theta_t(x) = -cx + \begin{cases} H_t(S_t^*) + K, & \text{if } x \le s_t^* \\ H_t(x), & \text{if } x > s_t^*. \end{cases} \tag{4.94}$$

Figure 4.8 sketches $\theta_t(x)$ and its constituent parts. The piecewise nature of $\theta_t(x)$ makes it nonconvex, even if $H_t(y)$ is convex. Figure 4.9 plots $\theta_t(x)$ for $t = 8, \ldots, 11$ for an instance with $T = 10$ and $c = 1$, $K = 100$, $h = h_T = 1$, $p = p_T = 5$, $\gamma = 1$, $\mu = 100$, and $\sigma = 10$.

Fortunately, although we used convexity to prove optimality of an (s, S) policy in the single-period case, convexity is not required—an (s, S) policy is still optimal under a weaker condition.

Let $f(x)$ be a real-valued function and let $K \ge 0$. Then, f is *K-convex* if, for all x and all $a, b > 0$,

$$f(x) + a \cdot \frac{f(x) - f(x-b)}{b} \le f(x+a) + K \tag{4.95}$$

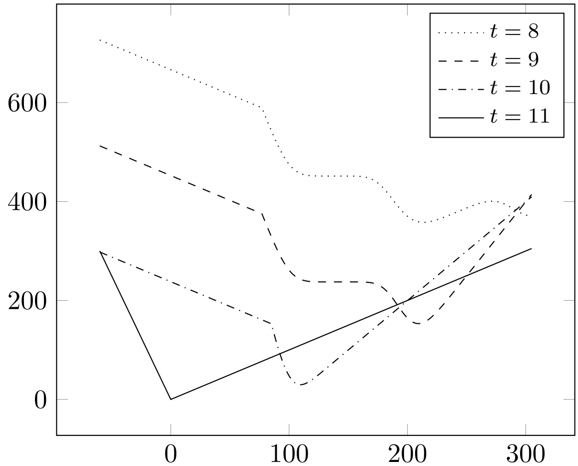


Figure 4.9 $\theta_t(x)$ for $t = 8, \ldots, 11$; $c = 1$, $K = 100$, $h = h_T = 1$, $p = p_T = 5$, $\gamma = 1$, $\mu = 100$, $\sigma = 10$, $T = 10$.

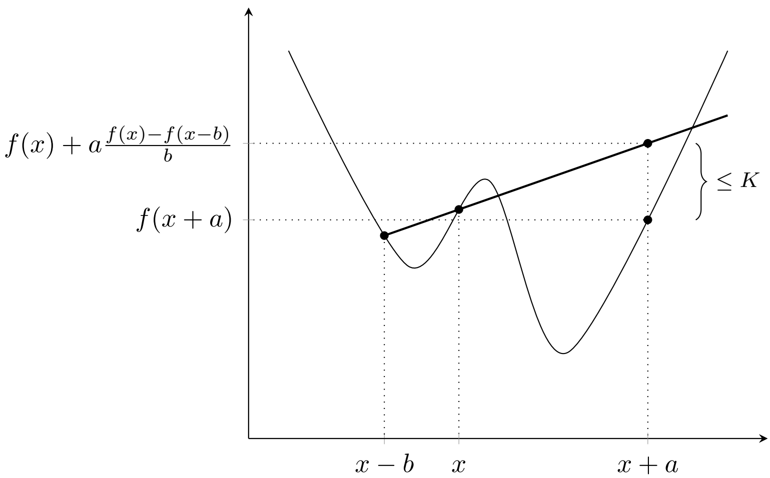


Figure 4.10 K-convexity.

(Scarf 1960). This definition is identical to (one) definition of convexity, except for the $+K$ on the right-hand side. The term $[f(x) - f(x-b)]/b$ is similar to a derivative at x (think about b approaching 0). Then the left-hand side of (4.95) approximates $f(x+a)$ by linearizing it using the "slope" of f between $x-b$ and x. (See Figure 4.10.) Therefore, K-convexity implies that this approximation doesn't overestimate $f(x+a)$ by more than K. (It may also underestimate it.) If f is convex, then the approximation on the left-hand side of (4.95) *always* underestimates $f(x+a)$. That is, (4.95) holds with $K = 0$. Therefore, 0-*convexity is equivalent to convexity*.

It is worth noting that, whereas some other convexity-like properties that you may be familiar with—quasiconvexity, pseudoconvexity, and so on—are used outside of inventory

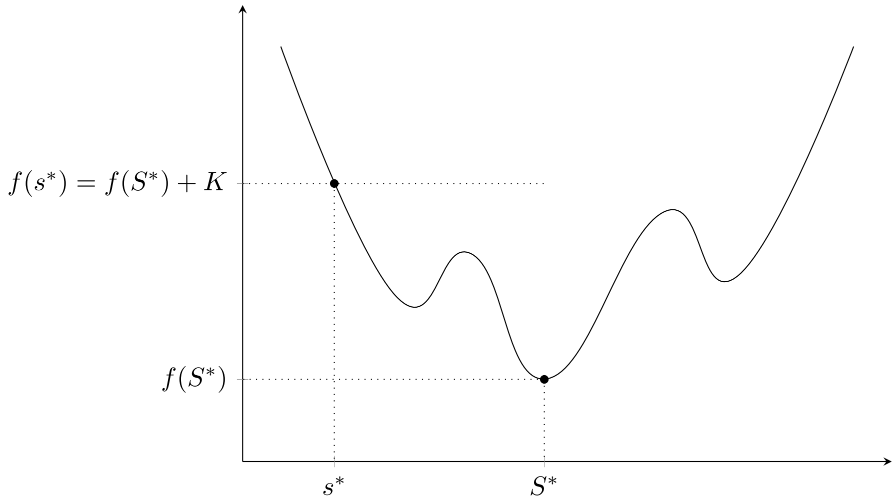


Figure 4.11 Properties of K-convex functions from Lemma 4.16.

theory, K-convexity was developed specifically for proving the optimality of (s, S) policies and (as far as we know) is not used outside of inventory theory.

Here is another important property of K-convexity:

Lemma 4.16 *Let f be a continuous, K-convex function. Let S^* be its smallest global minimizer and let s^* be the largest $x \leq S^*$ such that $f(x) = f(S^*) + K$. Then:*

(a) f is nonincreasing on $(-\infty, s^*]$.

(b) If $s^* < x \leq S^*$, then $f(x) < f(s^*)$.

(c) Suppose $S^* < x_1 < x_2$. Then $f(x_1) - f(x_2) \leq K$.

Lemma 4.16 says that a K-convex function first decreases for a while, up to a point s^*; then, after a different point S^*, if it ever decreases, it never decreases by more than K; and, in between these two points, the function never rises above its value at s^*. (See Figure 4.11.) This property will lead to the optimality of an (s, S) policy (as you may have suspected from our choice of notation in the lemma).

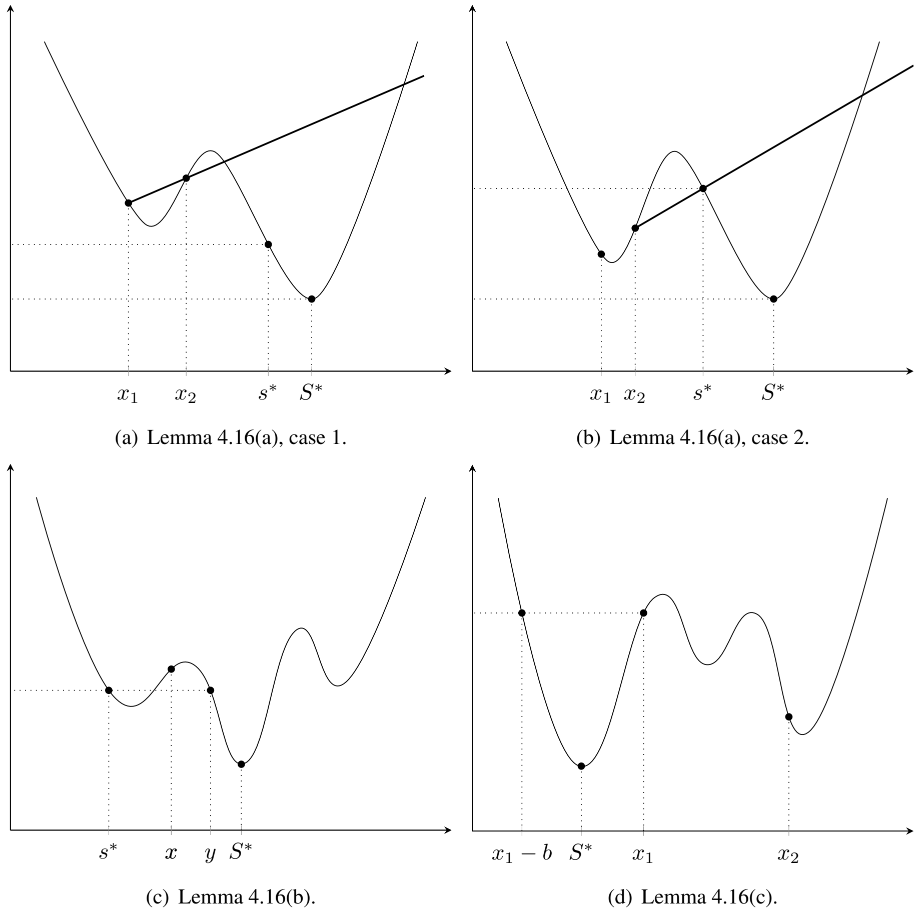
Proof.

(a) Suppose (for a contradiction) that f is *not* nonincreasing on $(-\infty, s^*]$. Then there exists $x_1 < x_2 < s^*$ such that $f(x_1) < f(x_2)$. We consider two cases.

Case 1: $f(x_2) \geq f(s^*)$. (See Figure 4.12(a).)

Let $b = x_2 - x_1$ and $a = S^* - x_2$. Then,

$$\begin{aligned}
f(x_2) + a \cdot \frac{f(x_2) - f(x_2 - b)}{b} &= f(x_2) + (S^* - x_2)\frac{f(x_2) - f(x_1)}{b} \\
&> f(x_2) \quad \text{(since } f(x_2) - f(x_1) > 0\text{)} \\
&\geq f(s^*) \quad \text{(by case 1 assumption)} \\
&= f(S^*) + K
\end{aligned}$$



(a) Lemma 4.16(a), case 1. (b) Lemma 4.16(a), case 2.

(c) Lemma 4.16(b). (d) Lemma 4.16(c).

Figure 4.12 Proof of Lemma 4.16.

$$= f(x_2 + a) + K.$$

This contradicts the K-convexity of f.

Case 2: $f(x_2) < f(s^*)$. (See Figure 4.12(b).)

Let $b = s^* - x_2$ and $a = S^* - s^*$. Then

$$\begin{aligned} f(s^*) + a \cdot \frac{f(s^*) - f(s^* - b)}{b} &= f(s^*) + (S^* - s^*)\frac{f(s^*) - f(x_2)}{b} \\ &> f(s^*) \quad (\text{since } f(s^*) - f(x_2) > 0) \\ &= f(S^*) + K \\ &= f(s^* + a) + K. \end{aligned}$$

This contradicts the K-convexity of f.

Since both cases lead to a contradiction, f must be nonincreasing on $(-\infty, s^*]$.

(b) Let $s^* < x \le S^*$. Suppose (for a contradiction) that $f(x) \ge f(s^*)$. (See Figure 4.12(c).) Then, by the continuity of $f(\cdot)$, there is some y, $x \le y < S^*$, such that $f(y) = f(s^*) = f(S^*) + K$, which violates the definition of s^* as the *largest* $x \le S^*$ such that $f(x) = f(S^*) + K$. Therefore, $f(x) < f(s^*)$.

(c) Suppose (for a contradiction) that there exists some x_1 and x_2 such that $S^* < x_1 < x_2$ but $f(x_1) - f(x_2) > K$. (See Figure 4.12(d).)

Let b be defined such that $f(x_1 - b) = f(x_1)$. (We'll assume such a b exists. It does if $\lim_{x \to -\infty} f(x) = \infty$, which is true of the K-convex functions we'll consider below.) Let $a = x_2 - x_1$. Then

$$\begin{aligned} f(x_1) + a \cdot \frac{f(x_1) - f(x_1 - b)}{b} &= f(x_1) \quad (\text{since } f(x_1 - b) = f(x_1)) \\ &> f(x_2) + K \quad (\text{by assumption}) \\ &= f(x_1 + a) + K. \end{aligned}$$

This contradicts the K-convexity of f. ■

The following properties of K-convex functions will be important in the results that follow. Parts (a)–(c) are generalizations of well-known results for convexity.

Lemma 4.17

(a) If $f(x)$ is K-convex, then $f(x + \epsilon)$ is K-convex for all constants ϵ.

(b) If $f_1(x)$ is K_1-convex and $f_2(x)$ is K_2-convex, then $\alpha_1 f_1(x) + \alpha_2 f_2(x)$ is $(\alpha_1 K_1 + \alpha_2 K_2)$-convex, for any $\alpha_1, \alpha_2 > 0$.

(c) If $f(x)$ is K-convex and Y is a random variable, then $\mathbb{E}_Y[f(x - Y)]$ is K-convex.

(d) If $f(x)$ is K_1-convex and $K_2 > K_1$, then $f(x)$ is K_2-convex.

Proof. Omitted; see Problem 4.42. ■

Now we're finally ready to prove the optimality of (s, S) policies for the finite-horizon problem. The logic will be similar to the base-stock case: The K-convexity of $\theta_{t+1}(x)$ implies the K-convexity of $H_t(y)$, which implies the optimality of an (s, S) policy in period t and the K-convexity of $\theta_t(x)$; and so on. The result was first proven by Scarf (1960); we follow the basic outline of his proof but use different arguments for some of the details.

Lemma 4.18 *If $\theta_{t+1}(x)$ is continuous and K-convex, then:*

(a) $H_t(y)$ is continuous and K-convex.

(b) An (s, S) policy is optimal in period t, with S_t^* equal to the smallest minimizer of $H_t(y)$ and s_t^* equal to the largest $x \le S_t^*$ such that $H_t(x) - H_t(S_t^*) = K$.

(c) $\theta_t(x)$ is continuous and K-convex.

Proof.

(a) We know that

$$H_t(y) = cy + g(y) + \gamma \mathbb{E}_D[\theta_{t+1}(y - D)].$$

The first two terms are each convex (i.e., 0-convex). Since $\theta_{t+1}(x)$ is K-convex (by assumption), $\mathbb{E}_D[\theta_{t+1}(y-D)]$ by Lemma 4.17(c), and $\gamma\mathbb{E}_D[\theta_{t+1}(y-D)]$ is K-convex by Lemma 4.17(b) and (d) since $\gamma \le 1$. Therefore, $H_t(y)$ is $(0+0+K)$-convex, or K-convex, by Lemma 4.17(b). Continuity follows from the continuity of each of the three terms.

(b) First note that Lemma 4.16 applies to $H_t(y)$ since it is K-convex and that the definitions of S_t^* and s_t^* are identical to those of S^* and s^* in the lemma. We'll determine the optimal ordering action for each starting inventory level x. If $x < s_t^*$, then by Lemma 4.16(a), $H_t(x) \ge H_t(s_t^*) = H_t(S_t^*)+K$, so it is cheaper to order up to S_t^* than not to order (and there is no better order-up-to level since S_t^* minimizes $H_t(y)$). If $s_t^* < x \le S_t^*$, then $H_t(x) < H_t(s_t^*)$ by Lemma 4.16(b). Therefore, $H_t(x) < H_t(S_t^*)+K$, so it is better to order nothing than to place an order. Finally, if $x > S_t^*$, then by Lemma 4.16(c), for any $y > x$, $f(x) < f(y)+K$, so it is better to order nothing than to place an order. This is exactly the definition of an (s,S) policy with parameters s_t^* and S_t^*.

(c) From (4.94), we know that

$$\theta_t(x) = -cx + \psi_t(x),$$

where

$$\psi_t(x) \equiv \begin{cases} H_t(S_t^*)+K, & \text{if } x \le s_t^* \\ H_t(x), & \text{if } x > s_t^*. \end{cases}$$

Clearly, each of the pieces of $\psi_t(x)$ is continuous, and at the breakpoint $x = s_t^*$, we have $H_t(S_t^*)+K = H_t(x)$ by the definition of s_t^* from part (b). Therefore, $\psi_t(x)$ is continuous, and so is $\theta_t(x)$.

To prove K-convexity, let x be any real number and let $a, b > 0$. Since $-cx$ is convex, it suffices to prove that $\psi_t(x)$ is K-convex. (Refer to Figure 4.8.)

If $x-b > s_t^*$, then $\psi_t(y) = H_t(y)$ for $y \in [x-b, x+a]$, so the K-convexity of ψ_t follows from that of H_t.

If $x+a \le s_t^*$, then $\psi_t(y) = H_t(S_t^*)+K$, a constant, for $y \in [x-b, x+a]$, so the K-convexity of ψ_t is trivial.

Suppose $x-b \le s_t^* < x+a$. We consider two cases. First, if $\psi_t(x) \le H_t(S_t^*)+K$, then

$$\begin{aligned}
&\psi_t(x) + a\cdot\frac{\psi_t(x)-\psi_t(x-b)}{b} \\
\le&\psi_t(x) \quad (\text{since } \psi_t(x) \le H_t(S_t^*)+K = \psi_t(x-b)) \\
\le&H_t(S_t^*)+K \\
\le&H_t(x+a)+K \quad (\text{since } S_t^* \text{ minimizes } H_t) \\
=&\psi_t(x+a)+K \quad (\text{since } x+a > s_t^*).
\end{aligned}$$

If, instead, $\psi_t(x) > H_t(S_t^*)+K$, then $x > S_t^*$ and so $\psi_t(x) = H_t(x)$. Then

$$\psi_t(x) + a\cdot\frac{\psi_t(x)-\psi_t(x-b)}{b}$$

$$
\begin{aligned}
&= H_t(x) + a \cdot \frac{H_t(x) - (H_t(S_t^*) + K)}{b} \\
&\le H_t(x) + a \cdot \frac{H_t(x) - H_t(S_t^*)}{x - S_t^*} \quad \text{(since } K \ge 0 \text{ and } x - b \le s_t^* \le S_t^*\text{)} \\
&\le H_t(x+a) + K \quad \text{(by } K\text{-convexity of } H_t\text{, letting } b' = x - S_t^*\text{)} \\
&= \psi_t(x+a) + K \quad \text{(since } x + a > s_t^*\text{)}.
\end{aligned}
$$

Therefore, $\psi_t(x)$ is K-convex, and so is $\theta_t(x)$. ■

Theorem 4.19 *If the terminal cost function $\theta_{T+1}(x)$ is continuous and convex, then an (s,S) policy is optimal in each period of the finite-horizon problem with fixed costs.*

Proof. By assumption, $\theta_{T+1}(x)$ is continuous and convex. Therefore, by Lemma 4.18(b), an (s,S) policy is optimal in period T. Moreover, $\theta_T(x)$ is continuous and K-convex by Lemma 4.18(c). This implies that an (s,S) policy is optimal in period $T-1$ and that $\theta_{T-1}(x)$ is continuous and K-convex. Continuing this logic, an (s,S) policy is optimal in every period. ■

4.5.2.3 Infinite Horizon If $T = \infty$, it is still true that an (s,S) policy is optimal in every period. And, echoing the infinite-horizon model with no fixed costs, the optimal s and S are the same in every period. However, the proof of these facts is quite a bit more difficult than the analogous proof in Section 4.5.1.3, and we omit it here. (See Zheng (1991).)

4.6 LOST SALES

Throughout this chapter, we have assumed that unmet demands are backordered. In this section, we assume instead that they are lost. The distinction is only important when $T > 1$. (When $T = 1$, unmet demands can *only* be lost.)

4.6.1 Zero Lead Time

In this section, we assume that the lead time $L = 0$. First consider the case in which $K = 0$. In the finite-horizon model, the DP recursion (4.36) changes only slightly:

$$\theta_t(x) = \min_{y \ge x} \{c(y-x) + g(y) + \gamma \mathbb{E}_D[\theta_{t+1}\left((y-D)^+\right)]\}. \tag{4.96}$$

The only change is in the last term, where we take the positive part of $y - D$ to reflect the fact that the inventory level cannot become negative. A base-stock policy is still optimal (Problem 4.44), provided that the terminal cost function $\theta_{T+1}(x)$ is convex and nondecreasing. (Under backorders, we required convexity but not monotonicity, but monotonicity is usually not a restrictive assumption under lost sales. For example, one common terminal cost function under backorders, $\theta_{T+1}(x) = h_{T+1}x^+ + p_{T+1}x^-$, is *not* nondecreasing, but under lost sales, $x^- = 0$, and the resulting function, $\theta_{T+1}(x) = h_{t+1}x^+$ *is* nondecreasing.) The DP algorithm, Algorithm 4.1, applies without modification.

A base-stock policy is still optimal for the infinite-horizon model. Under the average-cost criterion ($\gamma = 1$) with lost sales, it is no longer true that we order μ items per period, on average, independent of the base-stock level; therefore, we must modify the expected cost function (4.38) to account for the purchase cost. In particular, with probability $1 - F(S)$, we end the previous period with $IL = 0$ and must order S units at the start of the current period; and otherwise, we must order the demand from the previous period. Therefore,

$$\begin{aligned}
g(S) =& c\left((1 - F(S))S + \int_0^S df(d)dd\right) \\
&+ h\int_0^S (S-d)f(d)dd + p\int_S^\infty (d-S)f(d)dd \\
=& cS + (h-c)\int_0^S (S-d)f(d)dd + p\int_S^\infty (d-S)f(d)dd \qquad (4.97)\\
=& cS + (h-c)\bar{n}(S) + pn(S). \qquad (4.98)
\end{aligned}$$

The first-order condition yields

$$S^* = F^{-1}\left(\frac{p-c}{h+p-c}\right).$$

The solution changes only slightly under the discounted-cost criterion:

$$S^* = F^{-1}\left(\frac{p-c}{h+p-\gamma c}\right). \qquad (4.99)$$

(In fact, (4.99) holds for the average-cost criterion, too, setting $\gamma = 1$.)

When $K \geq 0$, an (s, S) policy is still optimal (Veinott 1966). In the single-period problem, we set S^* and s^* as described in Section 4.4.2, *unless* s^* would be negative, in which case we set $s^* = 0$. The finite-horizon model (Section 4.4.3) can be modified in a manner similar to (4.96).

4.6.2 Nonzero Lead Time

Now we allow $L \geq 0$. Recall from Section 4.3.4.1 that under backorders, the infinite-horizon model with $K = 0$ extends easily to nonzero lead times. Unfortunately, the same is not true under lost sales. The reason is that the logic behind the conservation-of-flow equation (4.41) breaks down: We can no longer subtract the entire demand in periods $t, \ldots, t+L$ because a given demand only reduces the inventory level in period $t+L$ if the inventory level was sufficient when the demand occurred. The problem can be formulated as a DP, but with an L-dimensional state space. For reasonable values of L, the DP is typically impossible to solve exactly due to the curse of dimensionality. Many heuristics and approximations have been proposed; see, for example, Zipkin (2008a), Bijvank and Vis (2011), or Goldberg et al. (2016) for reviews.

A base-stock policy is no longer optimal (Karlin and Scarf 1958) for the nonzero-lead-time problem, and in fact the optimal policy form is unknown, aside from a few partial results about its structure—for example, that the optimal order quantity is decreasing in the on-hand inventory (Karlin and Scarf 1958) and that it is zero for certain vectors of on-order inventory (Morton 1969); Zipkin (2008b) proves these and other properties using the concept of $L^{\natural}$-convexity from discrete convex analysis.

On the other hand, Huh et al. (2009) prove that a base-stock policy is asymptotically optimal as $p/h \to \infty$, and Goldberg et al. (2016) prove the asymptotic optimality as $L \to \infty$ of an even simpler policy in which we order the same quantity in every period. Moreover, Levi et al. (2008) introduce a 2-approximation algorithm (i.e., a heuristic with a fixed worst-case error bound of 2). Their heuristic uses a *dual-balancing policy*, which means that it balances the expected marginal holding cost and the expected marginal stockout cost in each period. Order quantities in the dual-balancing policy can be computed much more efficiently than using DP. Chen et al. (2014) present a different approximation scheme, with an (additive) error bound that can be as small as the modeler likes (but with a corresponding increase in computational complexity).

Not surprisingly, when $K > 0$, the situation is even more complicated, and optimal policies are unknown for this case, too; see, e.g., Nahmias (1979).

Lost-sales problems with nonzero lead times are still, in many respects, an open problem and are an active area of research.

CASE STUDY 4.1 Optimization of Warranty Inventory at Hitachi

Hitachi is a global manufacturer of computer components, power grid equipment, construction vehicles, defense systems, and a wide range of other high-tech and heavy-duty products. In the early 2000s, they collaborated with researchers from Stanford University to optimize the inventory used to service warranties for disk drives. Khawam et al. (2007) discuss in detail the project, which we summarize here.

A customer who returns a defective drive may choose to receive a replacement or a credit for the value of the drive. The drives sent as replacements are usually remanufactured drives that were previously returned, and this project focused on managing the inventory of such remanufactured drives. When the inventory is depleted, the company must either purchase brand-new drives (which are more expensive than remanufactured ones) from the factory or make the customer endure excessive lead times. Although the warranty claims for hard drives follow a lifecycle curve similar to that of the product's demands, e.g., a Bass diffusion process (Section 2.6), the researchers chose to focus on the steady-state portion of a given product rather than the ramp-up or -down phases.

They modeled the warranty inventory system as a single-stage, periodic-review, infinite-horizon inventory system with backorders. Hitachi promises that warranty claims will be served within L_c periods. Their objective in this project was to determine the minimum inventory levels required to satisfy a type-2 service level constraint that required the percentage of replacements that are completed within L_c periods to be at least β. This ignores customers who prefer a credit instead of a replacement since credits can be processed very quickly.

A fraction δ of drives that are returned to Hitachi are tested to determine whether they are actually fully operational (called "no defects found," or NDF); the remaining $1 - \delta$ fraction are clearly defective and do not need testing. Of the drives sent for NDF testing, a fraction γ pass the NDF test and can be added to inventory, whereas the remaining $1 - \gamma$ of the drives are found to be defective. Defective drives (drives that fail the NDF test as well as those that did not undergo NDF testing) are sent for remanufacturing; a fraction θ of those are successfully remanufactured and added to

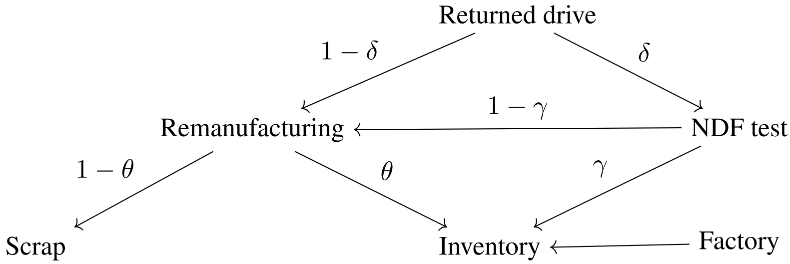


Figure 4.13 Process for handling returned drives at Hitachi.

inventory, while the remaining $1-\theta$ must be scrapped. Finally, the inventory manager can order new products from the manufacturing process, which incurs a lead time of L_m periods. The process is summarized in Figure 4.13.

Let $\eta_n = \gamma\delta$ be the fraction of returned units that pass the NDF test, $\eta_f = (1-\gamma)\delta\theta$ be the fraction that fail NDF but are successfully remanufactured, and $\eta_r = (1-\delta)\theta$ be the fraction that are not NDF tested but are successfully remanufactured. The total fraction of claims that cannot be satisfied from returns-based inventory and instead must be ordered from the factory is $(1-\alpha) - \eta_n - \eta_f - \eta_r$, since α fraction of the claims request a credit rather than replacement.

The inventory decisions that we must optimize are the replenishment orders placed to the factory. A base-stock policy is *not* optimal for this system. However, as no simple optimal policy is known, it is reasonable to assume the system uses a base-stock policy for the replenishment orders.

The demand for warranty claims in a given period is assumed to be $N(\mu, \sigma^2)$. The number of replacement drives demanded in a period—the "positive" demand—is therefore

$$D^+ \sim N\left((1-\alpha)\mu, (1-\alpha)\sigma^2\right).$$

On the other hand, returned units that are successfully added to inventory in a period can be considered as "negative" demand:

$$D^- \sim N\left(\sum_{i\in\{n,f,r\}} \eta_i\mu, \sum_{i\in\{n,f,r\}} \eta_i\sigma_i^2\right).$$

The net demand that must be satisfied from inventory in period t is the difference between the two: $D_n \sim N(\mu_n, \sigma_n^2)$, where

$$\mu_n = \left[(1-\alpha) - \sum_{i\in\{n,f,r\}} \eta_i\right]\mu$$

$$\sigma_n^2 = \left[(1-\alpha) - \sum_{i\in\{n,f,r\}} \eta_i\right]\sigma^2.$$

(We assume the term inside the $[\cdot]$ is positive, otherwise the supply exceeds the demand.)

The L_c-period lead time that Hitachi promises its customers for replacement drives in effect reduces the supply lead time of L_m periods. That is, the lead-time demand should be interpreted as the demand over $L_m - L_c + 1$ periods rather than over $L_m + 1$ periods. This is a *net lead time*, which we discuss in detail in Section 6.3; see also Hariharan and Zipkin (1995). The net-lead-time demand is normally distributed with mean $(L_m - L_c + 1)\mu_n$ and variance $(L_m - L_c + 1)\sigma_n^2$.

We want to ensure a type-2 service level of β. Using $\hat{B}_1$ from (4.58), we have

$$n^{L_m - L_c + 1}(S) = (1 - \beta)(1 - \alpha)\mu, \tag{4.100}$$

where $n^{L_m - L_c + 1}(\cdot)$ is the loss function for the net-lead-time demand. Equation (4.100) can be solved numerically to find S.

The researchers also developed models with (1) random rather than deterministic yields of the processes in Figure 4.13, using ideas similar to those in Section 9.3; (2) random lead times in the remanufacturing process, using formulas similar to (5.24) and (5.25); and (3) order batching, using a model similar to that in Section 13.2.4. They embedded these models into a spreadsheet, into which planners could input the weekly demand forecast (using a moving average) and other parameters. The model outputs included the base-stock level and the resulting average inventory level, expressed in terms of weeks of supply (inventory units divided by units demanded per week). Interestingly, the optimal base-stock levels for different product families were very close to each other when expressed in weeks of supply, even though the input parameters differed considerably.

The research team rolled out the spreadsheet tool to planners, who used it as part of the planning process for warranty servicing. The tool—and the process of developing it—was also valuable to planners for learning more about the operation of the inventory system.

PROBLEMS

4.1 (Inventory of Ski Jackets) A clothing company sells ski jackets every winter but must decide in the summer how many jackets to produce. Each jacket costs \$65 to produce and ship and sells for \$129 at retail stores. (For the sake of simplicity, assume the jacket is sold in a single store.) Customers who wish to buy this jacket but find it out of stock will buy a competitor's jacket; in addition to the lost revenue, the company also incurs a loss-of-goodwill cost of \$15 for each lost sale. At the end of the winter, unsold jackets are sold to a discount clothing store for \$22 each.

- **a)** First suppose that the demand for the ski jackets this winter will be distributed as a normal random variable with mean 900 and standard deviation 60. What is the optimal number of jackets to produce?
- **b)** Now suppose that the demand is distributed as a Poisson random variable with mean 900. What is the optimal number of jackets to produce?

4.2 (Dixie's Stew) One of the specialties at Dixie's Cafe is vegetable stew, which simmers over a low flame all day. Since the cooking time is so long, Dixie must decide in the morning how many servings of the stew to cook for that night's dinner service. Moreover, the stew cooked on a given day cannot be served the next day; it must be thrown away. Vegetable stew is the highest-profit item on the menu at Dixie's Cafe. It earns Dixie a profit of \$8 per

Table 4.3 Demand for in-flight meals for Problem 4.3.

d	Probability $f(d)$	Cumulative Probability $F(d)$
40	0.01	0.01
41	0.03	0.04
42	0.04	0.08
43	0.05	0.13
44	0.08	0.21
45	0.09	0.3
46	0.12	0.42
47	0.13	0.55
48	0.17	0.72
49	0.12	0.84
50	0.08	0.92
51	0.03	0.95
52	0.02	0.97
53	0.02	0.99
54	0.01	1

serving, whereas all the other items earn a profit of \$4. Customers who want stew but find it out of stock will order one of these other items. The ingredients for one serving of stew cost the Cafe \$2.50.

a) First suppose that the demand for stew on a given evening is normally distributed with a mean of 18 and a variance of 16. How many servings of stew should Dixie prepare in the morning? (Fractional servings are OK.) What is the expected cost (ingredients and lost profit) of the optimal solution?

b) Now suppose that the demand is distributed as an exponential random variable with mean 18. How many servings should Dixie prepare?

4.3 (In-Flight Meals) Oceanic Airlines sells meals aboard their flights. Obviously, the airline must decide how many meals to put on the airplane before the flight takes off, and it cannot restock additional meals if it runs out during the flight. Each meal sells for \$7 and costs the airline \$2.50. If there are meals left over at the end of the flight, the perishable items must be thrown away, but nonperishable items (crackers, napkins, etc.) may be reused. The value of the reusable items is estimated at \$1.50. Assume there are no loss-of-goodwill penalties for unmet demand, only the lost profit.

a) Suppose the demand for meals on today's flight #815 has the distribution given in Table 4.3. How many meals should Oceanic stock on the flight?

b) Suppose instead that the demand for meals on flight #815 has a normal distribution with mean 50 and standard deviation 10. Now how many meals should Oceanic stock?

c) Calculate the optimal expected profit for meals sold on flight #815, still assuming demands are $N(50, 10^2)$.

4.4 (Chemical Manufacturing) A chemical manufacturer produces a certain chemical compound every Sunday, which it then sells to its customers on Monday through Saturday.

The company earns a revenue of \$80 per kg of the compound sold. Each kg manufactured costs the company \$40. If any of the compound goes unsold by Saturday night, it must be destroyed safely, at a cost of \$15 per kg. The total demand for the chemical compound throughout the week has a normal distribution with a mean of 260 kg and a standard deviation of 80 kg.

a) How much of the chemical compound should the company produce every Sunday?

b) What is the expected cost (including manufacturing cost, lost profit, and disposal cost) per week?

4.5 (Cheesy Blasters) A restaurant sells a snack food called Cheesy Blasters. Cheesy Blasters are essentially nonperishable, and since they are a specialty item, customers who experience stockouts are willing to wait until a future day, i.e., their demands are backordered. Daily demand for Cheesy Blasters is distributed as $N(28.3, 7.1^2)$. The restaurant orders the product from its supplier each morning. Unsold Blasters held in inventory overnight incur a holding cost of \$0.75 per item, and backorders incur a penalty of \$3.50 per item.

a) Calculate the optimal base-stock level and expected cost per day.

b) Assuming the restaurant uses the base-stock level from part (a), calculate its type-1 and type-2 service levels. For type-2, calculate its exact service level, B, and both approximate service levels, $\hat{B}_1$ and $\hat{B}_2$.

c) Repeat part (b) assuming that the restaurant uses a base-stock level of 30.

d) Now assume that the restaurant can only place a replenishment order once per week (7 days), and that the supply lead time is 2 days. Calculate the optimal base-stock level and expected cost per period.

e) Repeat part (b) for the system described in part (d), using the optimal base-stock level.

4.6 (Electricity Generation) On day t, an electricity utility company must decide how much generation capacity to prepare for the electricity it will generate on day $t + 1$. Each megawatt-hour (MWh) of capacity prepared costs the utility r. Let S_{t+1} be the generation capacity chosen on day t for generation on day $t + 1$.

The demand for day $t + 1$, denoted D_{t+1}, is stochastic, with pdf $f(\cdot)$ and cdf $F(\cdot)$. D_{t+1} is not observed until day $t + 1$, although for simplicity we will assume that the entire day's demand is revealed at the beginning of the day.

Once D_{t+1} is observed, the utility generates $\min\{D_{t+1}, S_{t+1}\}$ MWh of electricity. Each MWh of electricity actually generated incurs a cost of c per MWh (in addition to the cost r already incurred to prepare the capacity). If $D_{t+1} > S_{t+1}$, the utility must purchase electricity on the *spot market* to make up the difference. (The spot market is a marketplace in which the utility can purchase an unlimited quantity of electricity with no advance notice required.) The price per MWh of electricity purchased on the spot market is m, with $m > r + c$.

a) Write an expression for S^*_{t+1}, the optimal number of MWh of capacity to prepare.

b) Suppose $r =$ \$5/MWh, $c =$ \$2/MWh, $m =$ \$20/MWh, and $D_{t+1} \sim N(150, 20^2)$ MWh. What is S^*_{t+1}?

4.7 (Newsvendor Applications #1) Each of the situations below can be interpreted as a newsvendor problem. For each, indicate the holding and stockout costs, h and p, and use the results of Section 4.3.2 to find S^*.

Table 4.4 Probability distribution of TV show duration for Problem 4.7(b).

x	$\mathbb{P}$(show lasts for x seasons)
1	0.25
2	0.05
3	0.10
4	0.20
5	0.15
6	0.10
7	0.10
8	0.05

a) You are about to sign a 2-year contract for a mobile phone and you need to decide how many minutes per month to commit to purchasing. You can purchase any number S of minutes. (You are not restricted to rate plans specified by your mobile phone company.) If you commit to purchasing S minutes per month, you pay \$0.05 for each of these S minutes (regardless of whether or not you use them), plus \$0.25 for each minute you use in excess of S.

(For example, if $S = 100$ and you use 120 minutes, you pay $100 \times 0.05 + 20 \times 0.25 = 10$.)

Your monthly usage of minutes has a normal distribution with mean 1000 and standard deviation 220.

b) You are the producer of a new TV show and are about to negotiate a contract with the star of the show. You need to decide how many seasons (years) to commit to in the contract, but you are not sure how many seasons of the show will be produced before it is canceled. For each season you commit to in the contract, the star's salary will be \$1.5 million. If you commit to S years but the show lasts for longer than that, you will have to pay the star \$2.5 million per season (since she will become more popular in the future and will demand a higher salary). If you commit to S years but the show is canceled earlier than that, you do not need to pay the star's salary for seasons that were not produced; instead, you must pay her a \$500,000 contract-cancellation fee for each season committed to but not produced.

(For example, if you commit to 3 seasons and the show is produced for 4 seasons, you will pay $\$1.5 \times 3 + \$2.5 \times 1 = \$7$ million. If the show is produced for 2 seasons, you will pay $\$1.5 \times 2 + \$0.5 \times 1 = \$3.5$ million.)

Table 4.4 lists your estimates that the show will last for exactly x seasons, for various values of x.

c) You are purchasing tickets for a group of students to attend a minor league baseball game. Tickets cost \$8 each when purchased in advance. The number of students who will actually show up to the game is random and has a Poisson distribution with mean 26. Suppose you purchase S tickets. If fewer than S students show up for the game, you can return the extra tickets to the box office for half of their original price. If more than S students show up for the game, you will need to buy tickets from "scalpers" (people selling tickets outside the stadium) for \$30 each.

4.8 (Newsvendor Applications #2) Follow the instructions for Problem 4.7 for each of the following situations.

a) You are the manager of an auto-repair shop at which every car requires the entire day to repair. The shop does not accept appointments; customers arrive randomly. All customers arrive exactly when the shop opens in the morning.

If the number of auto mechanics on duty on a given day, S, is at least as large as the number of customers that arrive in the morning, all of the customers' cars will be repaired. If the number of customers exceeds S, however, the extra customers leave and get their car repaired at a competing shop across the street.

The number of customers arriving in a given day has a Poisson distribution with a mean of 18. Each car that is repaired earns the shop a profit of \$470, and each mechanic on duty costs the shop \$200 per day.

b) At the beginning of the academic year, you need to decide how many "dining dollars" to put on your university ID card. Dining dollars earn you a 15% discount on the food you buy on campus—so \$100 in dining dollars buys you $100/0.85 = \$117.65$ in food. However, any dining dollars not spent by the end of the academic year are lost. (Yes—you could just stock up on soda and potato chips at the end of the year to spend your remaining dollars. But pretend that's not possible.) The (undiscounted) value of the food you buy in 1 year is given by the random variable X, which has a lognormal distribution with parameters $\mu = 6$ and scale parameter $\sigma = 0.3$. (That is, $\ln X$ has a normal distribution with mean 6 and standard deviation 0.3.)

c) A small cement manufacturer operates a single truck, which makes deliveries throughout the day. The company must decide how much cement to load onto the truck each morning, before knowing how much cement each customer will request. It costs the company \$20 per cubic yard loaded onto the truck, in materials and labor costs. For each cubic yard of cement sold, the company earns \$65 in profit. The total demand for cement in a given day (summed over all the firm's customers) is normally distributed with a mean of 7 cubic yards and a standard deviation of 3 cubic yards.

There is no opportunity to load more cement for the rest of the day. Any unused cement at the end of the day must be discarded, with no salvage value—in fact, it *costs* the company \$35 per cubic yard in labor to clean out the dried-up cement from the truck. Assume the truck's capacity is large enough to hold any desired amount of cement.

4.9 (Simulation of Mobile-Phone Contract) Simulate the system in Problem 4.7(a) for 1000 months in a spreadsheet program. For each month, generate a random variate from the appropriate distribution and calculate the resulting cost. In your writeup, include the first 10 rows of your spreadsheet and report the average total cost per month (including the cost of the contracted minutes).

4.10 (Simulation of Dining Dollars) Simulate the system in Problem 4.8(b) for 1000 years in a spreadsheet program. For each year, generate a random variate from the appropriate distribution and calculate the resulting cost. In your writeup, include the first 10 rows of your spreadsheet and report the average total overage and underage cost per year.

4.11 (Managing Blood Inventory) A hospital purchases blood from a local blood-donation organization and uses it for patients during surgeries and emergency procedures. The hospital pays \$175 for each unit of blood purchased. Orders must be placed first thing in the morning, and any blood not used by the end of the day must be discarded. There is no salvage value or cost to discard a unit of blood. If the hospital needs more blood on a given day than they purchased that morning, they must place an emergency order; blood ordered this way costs \$420 instead of \$175. The number of units of blood that the hospital uses on a given day is normally distributed, with a mean of 150 and a standard deviation of 40.

a) Interpret this problem as a newsvendor problem. What are the holding and stockout costs, h and p?

b) What is the optimal number of units of blood for the hospital to purchase in the morning? (Fractional answers are OK.)

c) On what fraction of days will the hospital need to order at least one emergency unit of blood?

d) Suppose unused inventory *costs* the hospital money to dispose. Will the optimal order quantity increase, decrease, or stay the same? Will the optimal expected cost increase, decrease, or stay the same?

4.12 (Inventory Simulation) Using a spreadsheet software package of your choice, simulate an infinite-horizon base-stock policy (Section 4.3.4). Your spreadsheet should include columns for the starting and ending inventory level; the order quantity; the random demand; and the total cost (as well as any other columns you wish to include). Use the optimal base-stock level S (which should be calculated within your spreadsheet) and assume that the system begins period 1 with S units on hand.

a) Assume that demands per period are $N(100, 20^2)$ and that $h = 3, p = 25, \gamma = 1$, and $L = 0$. Simulate the system for at least 1000 periods and include the first 10 rows of your spreadsheet in your report.

b) For each performance measure listed below, calculate the exact mean value (using formulas contained in this chapter) and the mean value from the simulation, and compare the two.

- Ending inventory level
- Order quantity
- Holding cost per period
- Stockout cost per period
- Total cost per period
- Type-1 service level
- Type-2 service level

4.13 (Inventory Simulation: Fixed Cost) Add a fixed ordering cost K to your simulation from Problem 4.12 and implement an (s, S) inventory policy. Calculate optimal, or near-optimal, values of the policy parameters s and S in the spreadsheet and use those for the simulation. Assume $K = 1000$. Report the simulated mean values for each of the performance measure listed in Problem 4.12(b). (Make sure to include the fixed cost when you report the total cost.)

4.14 (Inventory Simulation: Lead Time) Modify your simulation from Problem 4.12 to handle a nonzero lead time L. Calculate the optimal value of S in the spreadsheet and use it for the simulation. Assume $L = 4$. Report the simulated mean values for each of the performance measures listed in Problem 4.12(b).

4.15 (Implicit vs. Explicit Newsvendor Cost Functions) Let $h' = h + c - v$ and $p' = p + r - c$. Prove that the (implicit) newsvendor cost function (4.12) under cost parameters h' and p' is equal to the explicit newsvendor cost function (4.19) plus the constant $(r - c)\mu$, which represents the expected margin earned on the units sold.

4.16 (Discrete Newsvendor with Continuous Demands) Suppose that the newsvendor's demand has a continuous distribution but the newsvendor must choose integer values of S. Prove (by giving examples) that S^* can equal *either* $S - 1$ or S, where S is such that $F(S-1) < p/(h+p) < F(S)$.

4.17 (Alternate Fill Rate Formula) Silver and Bischak (2011) prove the following formula for the type-2 service level under an infinite-horizon base-stock policy with lead time $L \geq 0$ and reorder interval $R \geq 1$:

$$B = 1 - \left[\frac{\sqrt{L+R}}{R} \mathrm{CV} \mathscr{L}(z) - \frac{\sqrt{L}}{R} \mathrm{CV} \mathscr{L} \left(\frac{R}{\sqrt{L}\mathrm{CV}} + z\sqrt{\frac{L+R}{R}} \right) \right], \tag{4.101}$$

where $\mathrm{CV} = \sigma/\mu$ is the coefficient of variation for the demand in one period and

$$z = \frac{S - (L+R)\mu}{\sqrt{L+R}\sigma}.$$

Prove that (4.101) is equivalent to (4.55).

4.18 (Newsvendor with Forecasting) Suppose that demands are normally distributed and that the newsvendor does not know μ and σ, but he estimates them in each period, as described in Section 4.3.2.7, using moving averages and standard deviations with $N = 5$. The observed demands in periods $t - 10, \ldots, t - 1$ are 99, 87, 125, 106, 100, 107, 93, 114, 87, and 85. The cost parameters are $h = 2$ and $p = 15$. What is the optimal order quantity for the newsvendor in period t?

4.19 (Lognormal Newsvendor) Suppose the demand D has a lognormal distribution with parameters μ and σ. (That is, $\ln D \sim N(\mu, \sigma^2)$.) Prove that the optimal solution to the newsvendor problem and its expected cost are given by

$$\begin{aligned} S^* &= e^{\mu + z_\alpha \sigma} \\ g(S^*) &= (h+p)\mathbb{E}[D]\Phi(\sigma - z_\alpha) - h\mathbb{E}[D], \end{aligned}$$

where $\alpha = p/(h+p)$.

Hint: The loss function for the lognormal distribution for $x > 0$ is

$$n(x) = e^{\mu + \frac{\sigma^2}{2}} \Phi\left(\frac{\mu + \sigma^2 - \ln x}{\sigma} \right) - x\left(1 - \Phi\left(\frac{\ln x - \mu}{\sigma} \right) \right). \tag{4.102}$$

4.20 (The Cooperative Newsvendor) Consider a newsvendor who purchases newspapers from his supplier at a cost of c per newspaper and sells them at a price of r per newspaper. If he has unsold newspapers at the end of the day, he can take them to the local recycling center, which pays him a salvage value of v per newspaper. The daily demand for newspapers has pdf $f(x)$ and cdf $F(x)$. Assume that $F(x)$ is strictly increasing.

a) Write the newsvendor's expected cost as a function of S, denoted $g_n(S)$. (Your expression may include integrals.) Show that the order quantity that minimizes $g_n(S)$ is

$$S_n^* = F^{-1}\left(\frac{r-c}{r-v}\right).$$

b) Suppose the newsvendor's supplier prints newspapers on demand; that is, she observes the newsvendor's order of S and then prints exactly S newspapers. The supplier therefore faces no uncertainty. It costs the supplier b to print one newspaper. Write the supplier's expected net cost (i.e., cost minus revenue) as a function of S, denoted $g_s(S)$. Then write the total supply chain expected cost as a function of S, denoted $g_t(S)$—that is, $g_t(S) = g_n(S) + g_s(S)$.

c) Find the order quantity S_t^* that minimizes $g_t(S)$. (If the supplier and the newsvendor were both owned by a single firm that sought to minimize its total costs, this is the order quantity it would pick.)

d) Prove that $S_n^* = S_t^*$ if and only if $c = b$—that is, if and only if the supplier earns zero profit on each newspaper she sells to the newsvendor.

e) Prove that $g_t(S_n^*) = g_t(S_t^*)$ if and only if $c = b$, and $g_t(S_n^*) > g_t(S_t^*)$ otherwise.

f) In a short paragraph, discuss the implications of the results you proved in this problem. What does it mean for two supply chain partners that are each attempting to minimize their own costs rather than minimizing the total supply chain cost?

4.21 **($g(S^*)$ for Poisson Newsvendor)** Suppose that in the newsvendor problem, the demand per period, D, has a Poisson distribution with mean λ. Suppose further that there exists an S^* such that $F(S) = p/(h+p)$. Prove that

$$g(S^*) = (h+p)f(S^*)\lambda.$$

4.22 **(Non-Standard-Normal Loss Function)** Prove equation (4.25) (also given in (C.31)).

4.23 **(Loss Function Derivatives)** Prove equations (4.13) and (4.14) (also given in (C.15) and (C.16)).

4.24 **(Uniform Loss Functions)** Derive expressions for the first- and second-order loss and complementary loss functions for the continuous $U[0, 1]$ distribution.

4.25 **(A Simple Revenue Management Problem)** An airplane has n seats in coach class. Two types of travelers will purchase tickets for a certain flight on a certain date: leisure travelers, who are willing to pay only the *discounted fare* r_d, and business travelers, who are willing to pay the *full fare* r_f ($r_f > r_d$). The airline knows that the number of leisure travelers requesting tickets for this flight will be greater than n for sure, while the number of business travelers requesting tickets is a random variable X with a given cdf $F(x)$.

Assume that the leisure travelers always purchase their tickets before the business travelers do. (In practice, this is roughly true, which is why airfares increase as the flight date gets closer.) The airline wishes to sell as many seats as possible to business travelers since they are willing to pay more. However, since the number of such travelers is random and these customers arrive near the date of the flight, a sensible strategy is for the airline to allocate a certain number of seats Q for full fares and the remainder, $n - Q$, for discount fares.

The discount fares are sold first: The first $n - Q$ customers requesting tickets will be charged r_d, and the remaining $\leq Q$ customers will be offered the full price r_f. Some of the customers being offered r_f will be leisure travelers; these travelers will decline to buy a ticket. Similarly, it is possible that some of the seats sold to leisure travelers for r_d could have been sold to business travelers who would have been willing to pay r_f.

a) Show that the problem of finding the optimal number of full-fare seats, Q, is equivalent to a newsvendor problem. What should be used in place of the holding and stockout costs h and p? What is the critical ratio? What is the optimality condition (analogous to (4.16))?

b) Suppose that demand for full-fare seats is normally distributed with a mean of 40 and a standard deviation of 18. There are $n = 100$ seats on the flight, and the fares are $r_d = \$189$ and $r_f = \$439$. What is the optimal number of full-fare seats? (Fractional solutions are OK.)

c) For each of the following situations, will the optimal Q increase, decrease, or stay the same? Will the optimal cost increase, decrease, or stay the same? Briefly explain your answers.

i. The full-fare tickets are fully refundable, and with some probability each business traveler will cancel his or her ticket at the last minute, too late for the airline to resell the newly vacant seat.

ii. A fraction of leisure travelers are willing to pay full fare if they arrive after the discount seats are sold out.

iii. Unsold seats may be sold at the very last minute for a steeply discounted price (for example, on a discount airfare website). These tickets are made available after most (though not necessarily all) of the business travelers have requested tickets.

4.26 (Allocating Parking Spots) You are the manager of a luxury apartment building whose parking garage contains 300 parking spots. Residents may choose to purchase a *dedicated* parking spot for \$60,000 for 3 years. (Only 3-year contracts are available.) The garage also has *metered* parking spots that require drivers to pay \$4 per hour for parking. The number of drivers wishing to park in metered spots in a given hour has a normal distribution with a mean of 50 and a standard deviation of 10. Your goal is to choose how to allocate the 300 spots between dedicated and metered spots.

To keep things simple, assume that (1) the demand for dedicated spots is greater than 300; (2) drivers who park in metered spots all park for exactly 1 hour, arriving and departing on the hour (at 12:00, 1:00, etc.); and drivers who purchase dedicated spots never park in metered spots, and vice-versa.

What is the optimal number of spots to designate as metered spots?

4.27 (Free Overage) Suppose that, in the newsvendor problem, we are allowed up to r units of overage for free before incurring holding costs, where $r \geq 0$ is a constant. That is, the cost if we order S units and have a demand of d is $g(S, d) = h((S-r)-d)^+ + p(d-S)^+$.

a) Write the optimality condition (analogous to (4.16)).

b) Apply this to the "dining dollars" example in Problem 4.8(b), assuming that $r = \$50$.

4.28 (DP Walkthrough) The demand for a given product in each period equals 2 with probability 0.2, 1 with probability 0.5, and 0 with probability 0.3. Holding and stockout costs per period are given by $h = 2$ and $p = 5$. The purchase cost is $c = 1$, and there is no fixed cost. The order-up-to level y in each period must be in $\{0, 1, 2\}$. The planning horizon is $T = 3$ periods, and the terminal cost at the end of the horizon is given by $\theta_4(x) = 4x^+ + 6x^-$. We begin period 1 with $x = 2$ units on hand. Using Algorithm 4.1, determine $y_t(x)$ for $t = 1, 2, 3$ and for each feasible value of x. Also determine the expected cost for the entire horizon (including the terminal cost), given that we begin the horizon with $x = 2$. Work through the algorithm by hand and show your work.

4.29 (Implementing Base-Stock DP) Consider the finite-horizon model with no fixed costs of Section 4.3.3.

a) Implement the DP model in any programming language you wish.

b) Suppose $T = 10$, $c = 1$, $h = 0.5$, $p = 10$, and $\gamma = 0.98$. Suppose the demand per period is distributed as $N(20, 5^2)$ and the terminal cost function is given by

$$\theta_{T+1}(x) = h_{T+1}x^+ + p_{T+1}x^-,$$

where $h_{T+1} = h$ and $p_{T+1} = p$. Using your DP, find $y_t(x)$ and $\theta_t(x)$ for $t = 1, \ldots, 10$ and $x = -10, \ldots, 40$. Report these in two separate tables. Also report the optimal base-stock level S_t^* for periods $t = 1, \ldots, 10$.

c) Plot $y_t(x)$ for $t = 5$.

4.30 (Implementing (s, S) DP) Consider the finite-horizon model with fixed costs of Section 4.4.3.

a) Implement the DP model in any programming language you wish.

b) Suppose $T = 10$, $c = 1$, $K = 40$, $h = 1$, $p = 25$, and $\gamma = 0.98$. Suppose the demand per period is distributed as $N(18, 3^2)$ and the terminal cost function is given by

$$\theta_{T+1}(x) = h_{T+1}x^+ + p_{T+1}x^-,$$

where $h_{T+1} = h$ and $p_{T+1} = p$. Using your DP, find $y_t(x)$ and $\theta_t(x)$ for $t = 1, \ldots, 10$ and $x = -10, \ldots, 40$. Report these in two separate tables. Also report the optimal parameters s_t^* and S_t^* for periods $t = 1, \ldots, 10$.

c) Plot $y_t(x)$ for $t = 5$.

4.31 ((s, S) for Refrigerators) Weekly demand for refrigerators at an appliance store has a Poisson distribution with a mean of 4. The holding and stockout cost for refrigerators at the store are $h = \$40$ and $p = \$125$ per week, respectively. Replenishment orders for refrigerators incur a fixed cost of $K = \$150$.

a) Suppose we set $(s, S) = (4, 10)$. What is the expected cost per week?

b) Using Algorithm 4.2, find the optimal parameters (s, S), and the corresponding optimal cost.

4.32 (Approximate (s, S) Policies) Consider an infinite-horizon instance in which the demand per period is normally distributed with a mean of 190 and a standard deviation of 48, and in which the costs are given by $K = 60$, $h = 2$, and $p = 36$. Determine approximate values for s and S:

a) Using the (r, Q) approximation.

b) Using the power approximation.

4.33 (Ordering Capacities) Suppose that an ordering capacity of b units is imposed in the finite-horizon model with no fixed costs of Section 4.3.3. Sketch a plot of $y_t(x)$ vs. x, analogous to Figure 4.4. (The exact numbers are not important; what is important is the shape of the curve.)

4.34 (DP for Ordering Capacities) Suppose that an ordering capacity of b units is imposed in the finite-horizon model with fixed costs of Section 4.4.3.

a) Explain how to modify the DP from Section 4.4.3 to account for the ordering capacity.

b) Implement your DP from part (a). Using your DP, find $y_t(x)$ and $\theta_t(x)$ for $t = 1, \ldots, 10$ and $x = -10, \ldots, 40$ for the instance described in Problem 4.30(b) using a capacity of $b = 10$. Report $y_t(x)$ and $\theta_t(x)$ in two separate tables.

4.35 (Nonoptimality of (s, S) Policies for Ordering Capacities) Suppose that an ordering capacity of b units is imposed in the finite-horizon model with fixed costs of Section 4.4.3. Prove, by providing a counter-example, that an (s, S) policy is *not* necessarily optimal in every period of the finite-horizon version of this problem. (The (s, S) policy is modified in this case: If $IP \leq s$, we order $\min\{S - IP, b\}$, and otherwise, we order nothing, where IP is the current inventory position.)

4.36 (K-Convexity Is Not a Necessary Condition) In Section 4.5.2.2, we proved that if $H_t(y)$ is continuous and K-convex, then an (s, S) policy is optimal in period t. However, K-convexity is not a necessary condition: An (s, S) policy can still be optimal in period t even if $H_t(y)$ is not K-convex. Sketch a graph of a function $H_t(y)$ that is not K-convex but for which an (s, S) policy is optimal. Explain clearly (a) why the function is not K-convex and (b) why an (s, S) policy is optimal.

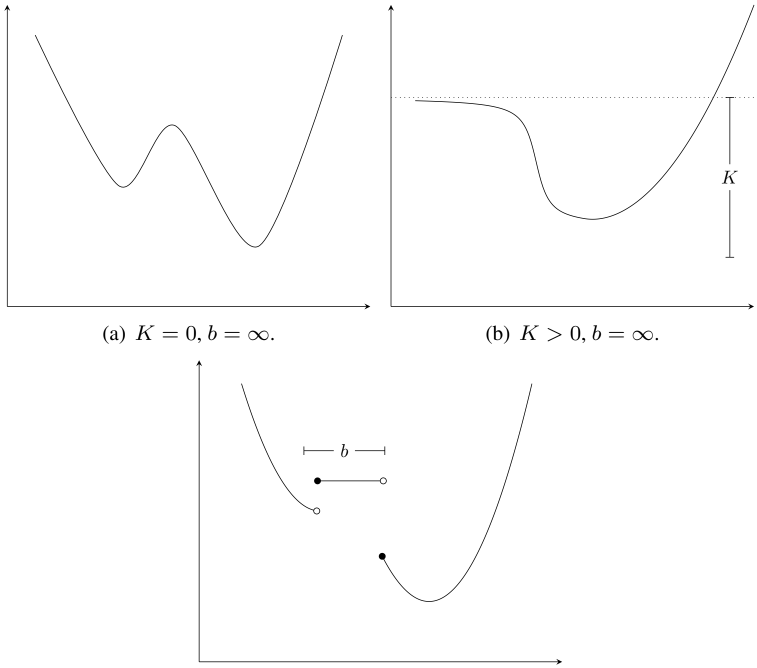
4.37 (Other Policy Forms #1) Consider the single-period model with no fixed costs from Section 4.5.1.1. We know that, for a given starting inventory level x, (4.83) determines the optimal inventory position after ordering, y. We assumed a particular form for $H(y)$ and used the convexity of this function to prove the optimality of a base-stock policy. But in principle $H(y)$ can have any form, and other policies may be optimal for other functions.

a) Develop a function $H(y)$ such that the optimal policy has three parameters, S_1, s_2, and S_2 ($S_1 < s_2 < S_2$), and has the following form:

- If $x \leq S_1$, then order up to S_1.
- If $S_1 < x \leq s_2$, do nothing.
- If $s_2 < x < S_2$, order up to S_2.
- If $x \geq S_2$, do nothing.

For the sake of simplicity, assume that $c = 0$. Sketch the function $H(y)$ and explain how to determine the optimal values of the parameters S_1, s_2, and S_2. (For example, "S_1 is the largest maximizer of $H(y)$.")

b) Now suppose that $K > 0$ so that the term $K\delta(y - x)$ is now added to the objective function, as in (4.92). Develop a function $H(y)$ such that the optimal policy has four parameters, s_1, S_1, s_2, and S_2 ($s_1 < S_1 < s_2 < S_2$), and has the following form:



(a) $K = 0, b = \infty$.

(b) $K > 0, b = \infty$.

(c) $K = 0, b < \infty$.

Figure 4.14 $H(y)$ functions for Problem 4.38, with fixed cost $K \geq 0$ and ordering capacity $b \leq \infty$.

- If $x \leq s_1$, then order up to S_1.
- If $s_1 < x \leq s_2$, do nothing.
- If $s_2 < x < S_2$, order up to S_2.
- If $x \geq S_2$, do nothing.

Sketch the function $H(y)$ and explain how to determine the optimal values of the parameters s_1, S_1, s_2, and S_2.

4.38 (Other Policy Forms #2) Describe the form of the optimal single-period inventory policy for each of the functions $H(y)$ depicted in Figure 4.14 (in a manner similar to the descriptions in Problem 4.37). Explain how to determine the optimal values of the parameters for your policy. Note that in part (c), there is a fixed cost of K, and in part (c), there is an ordering capacity of b units. For all parts, assume that the per-unit cost $c = 0$.

4.39 (Other Policy Forms #3) Suppose that, in the single-period model with fixed costs of Section 4.5.2.1, the function $H(y)$ has a shape similar to the curve in Figure 4.15, with $K > 0$.

a) Prove that $H(y)$ is not K-convex.

b) Describe the form of the optimal inventory policy (in a manner similar to the descriptions in Problem 4.37). Explain how to determine the optimal values of the parameters for your policy.

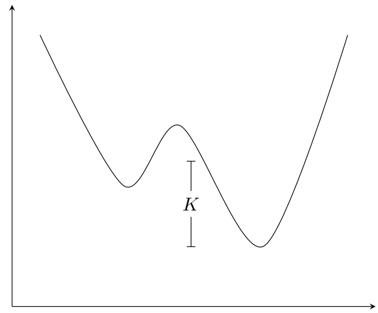


Figure 4.15 $H(y)$ function for Problem 4.39, with fixed cost $K > 0$.

c) Write a set of conditions on $H(y)$ that ensures that, if all of your conditions hold, then the policy that you described in part (b) is optimal. Your conditions must be sufficient but need not be necessary.

4.40 (Single-Period Control-Band Policies) Consider the single-period model without fixed costs of Section 4.5.1.1, and suppose we begin the period with an inventory level of $x \geq 0$. Suppose further that we can return excess inventory to the supplier in each period. That is, we can choose $Q < 0$, or equivalently, $y < x$.

For each unit we return, we earn a revenue of c', so the total revenue earned when $Q < 0$ is $-c'Q$. Normally $c' \geq 0$, but it's also possible that $c' < 0$, in which case we pay a *cost* to make the return.

Consider the following policy: There are two parameters, S and U, with $0 \leq S \leq U$. Set

$$y = \begin{cases} S, & \text{if } x < S \\ x, & \text{if } S \leq x \leq U \\ U, & \text{if } x > U. \end{cases}$$

The interval $[S, U]$ is called a *control band*, and the policy is called a *control-band policy*. The idea is to order up to S if x is below the control band, to "return down to" U if x is above the control band, and to do nothing if x is in the control band.

a) Prove that a control-band policy is optimal for the single-period problem.

b) Show how to calculate the optimal S^* and U^* for the single-period problem, and prove that $S^* \leq U^*$.

c) Prove that, in the single-period problem, as $c' \to -h$ (from above), $U^* \to \infty$. In a few sentences, explain why it is logical to require $c' \geq -h$.

d) Prove that, in the single-period problem, as $c' \to c$ (from below), $U^* - S^* \to 0$. In a few sentences, explain why it is logical to require $c' \leq c$.

e) Suppose the demand per period is distributed as $N(60, 12^2)$. Suppose $h = 0.4$, $p = 4.8$, $c = 3$, and $c' = 1.7$. Find S^* and U^* for the single-period problem.

4.41 (Finite-Horizon Control-Band Policies) Return to the setup in Problem 4.40, and now consider the finite-horizon model. Prove that a control-band policy is optimal in every period of the finite-horizon model. (The parameters of the control-band policy are now indexed by time, S_t and U_t.)

4.42 (Properties of K-Convex Functions) Prove Lemma 4.17.

4.43 (Alternate Terminal Cost Function) Consider the finite-horizon base-stock model described in Section 4.5.1.2. Suppose that the terminal cost function is given by

$$\theta_{T+1}(x) = \begin{cases} -(h+p)x, & \text{if } x \le 0 \\ 0, & \text{if } x > 0. \end{cases} \tag{4.103}$$

Suppose also that $h > \gamma c$.

a) Write an expression for $H_T(y)$.

b) Derive the optimal base-stock level in period T, in the form

$$S_T^* = F^{-1}([\text{some fraction}]).$$

c) Write an expression for $H_{T-1}(y)$. (*Note*: Your expression may involve cases, as in (4.90).)

d) Derive the optimal base-stock level in period $T-1$, in the form

$$S_{T-1}^* = F^{-1}([\text{some fraction}]).$$

e) Prove that $S_{T-1}^* < S_T^*$.

4.44 (Finite-Horizon Base-Stock Policies under Lost Sales) Prove that, if the terminal cost function $\theta_{T+1}(x)$ is convex and $\theta'_{T+1}(x) \ge -c$, then a base-stock policy is optimal in each period of the finite-horizon problem with no fixed costs under lost sales. (The condition $\theta'_{T+1}(x) \ge -c$ essentially ensures that the condition $p > c$ continues to hold even in the terminal cost function.)

4.45 (Minimum Order Quantity) Consider the single-period model without fixed costs from Sections 4.3.2 and 4.5.1. Suppose there is a constraint requiring the order quantity to be *either* 0 *or* at least M, where $M > 0$ is a constant.

a) One plausible policy for this problem is a modified base-stock policy in which we order $\max\{S - x, M\}$, where x is the starting inventory level. Prove (by giving a counterexample) that this policy is not optimal.

b) Another plausible policy is an (s, S) policy in which $S - x \ge M$. Prove that this policy is not optimal either.

c) Make a conjecture as to the form of the optimal policy. (That is, describe a decision rule, similar to how we described the policies in Section 4.1.)

d) *Bonus*: Specify the optimal parameters of the policy you described in part (c).

e) *Double Bonus*: Prove that the policy you described in parts (c) and (d) is optimal.

4.46 (Monotonic Safety Stock) Consider the infinite-horizon base-stock model with service-level constraints given by (4.62)–(4.64).

a) Suppose we use a type-1 service-level constraint (4.63). Argue that the optimal base-stock level and the optimal safety-stock level (given by the optimal base-stock level minus the mean lead-time demand, $S^* - (L+1)\mu$) both increase as the reorder interval R increases.

b) Suppose we use a type-2 service-level constraint (4.64) under approximation $\hat{B}_1$. Argue that the optimal base-stock level increases as R increases, but show that the optimal safety-stock level can *decrease* as R increases.

4.47 (Derivatives of $I(S)$ and $B(S)$) Assuming the demand is distributed $N(\mu, \sigma^2)$, prove that

$$\frac{\partial I}{\partial \sigma} = \frac{\partial B}{\partial \sigma} = \phi(z),$$

where $I(S)$ and $B(S)$ are as defined in Section 4.3.2.2 and $z = (S - \mu)/\sigma$.

4.48 (DP for New and Used Items) A company manufactures and sells a laptop computer that has a market both for new items and for used ones. In each period, the firm decides how much to manufacture and then observes the demand for each type (new and used). Demand is satisfied as much as possible, and then a portion of the unused new inventory "expires" and is considered used. Unmet demand for new products is backordered but unmet demand for used products is lost. Products cannot be substituted; that is, a customer demanding a used item cannot be given a new item, and vice-versa.

Use subscript 1 to denote new items and subscript 2 to denote used items. Thus, the holding and stockout costs per item per period for new items are given by h_1 and p_1, respectively. For used items, the holding cost per item per period is given by h_2, and the stockout cost per item is given by p_2. Assume that demands of type i ($i = 1, 2$) are independent and normally distributed with pdf $f_i(d)$ and that the demand for each type in a given period is independent of the demand for the other type. There is no fixed ordering cost, and the discount factor is γ.

The sequence of events in each time period is as follows:

1. The inventory levels IL_1 and IL_2 of new and used items (respectively) are observed.
2. A manufacturing order for new items is placed and is ready instantaneously.
3. Demands d_1 and d_2 for new and used items (respectively) are observed. As much demand as possible is satisfied from the two inventories. Unmet demands for new items are backordered and unmet demands for used items are lost.
4. $\beta IL_1'$ new items are transferred to the used inventory, where β is a constant ($0 \leq \beta \leq 1$) and IL_1' is the inventory level of new items after the manufacturing order is received and the demand is subtracted, i.e., after step 3.
5. Holding and stockout costs are assessed based on the ending on-hand inventory levels.

Let $\theta_t(x_1, x_2)$ be the optimal expected cost in periods $t, t+1, \ldots, T$ if we begin period t with a new-item inventory level of x_1 and a used-item inventory level of x_2 (and act optimally thereafter). Formulate a recursive (DP) expression for $\theta_t(x_1, x_2)$, analogous to (4.36).

Your expression must use y, the order-up-to level for new items, as the decision variable for the minimization. Do not write the expectation as $\mathbb{E}[\cdot]$. Instead, write out the expectation using integrals. If you define any additional notation, define it clearly.

CHAPTER 5

STOCHASTIC INVENTORY MODELS: CONTINUOUS REVIEW

5.1 (r, Q) POLICIES

In this chapter, we consider a setting similar to the economic order quantity (EOQ) model (Section 3.2) but with stochastic demand. The mean demand per year is λ. The inventory position is monitored continuously, and orders may be placed at any time. There is a deterministic lead time L (≥ 0). Unmet demands are backordered.

If the demand has a continuous distribution, then the inventory level decreases smoothly but randomly over time, with rate λ, as in Figure 5.1. (Think of liquid draining out of a tank at a fluctuating rate.) This is the interpretation used in most of this chapter. Or demands may occur at discrete points in time (as customers arrive), for example, if the demand follows a Poisson process, as in Section 5.5.

We'll assume the firm follows an (r, Q) *policy*: When the inventory position reaches a certain point (call it r), we place an order of size Q. L years later, the order arrives. In the intervening time, the inventory on hand may have been sufficient to meet demand, or we may have stocked out. Note that the inventory level (solid line in Figure 5.1) and inventory position (dashed line) differ from each other during lead times but coincide otherwise. An (r, Q) policy is known to be optimal for the setting described here, although we will not prove this.

Whereas the EOQ model has a single decision variable Q, an (r, Q) policy has two decision variables: Q (the *order quantity*, sometimes called the *batch size*) and r (the

Fundamentals of Supply Chain Theory, Second Edition. Lawrence V. Snyder and Zuo-Jun Max Shen.

Companion website: www.wiley.com/go/Snyder/SupplyChainTheory

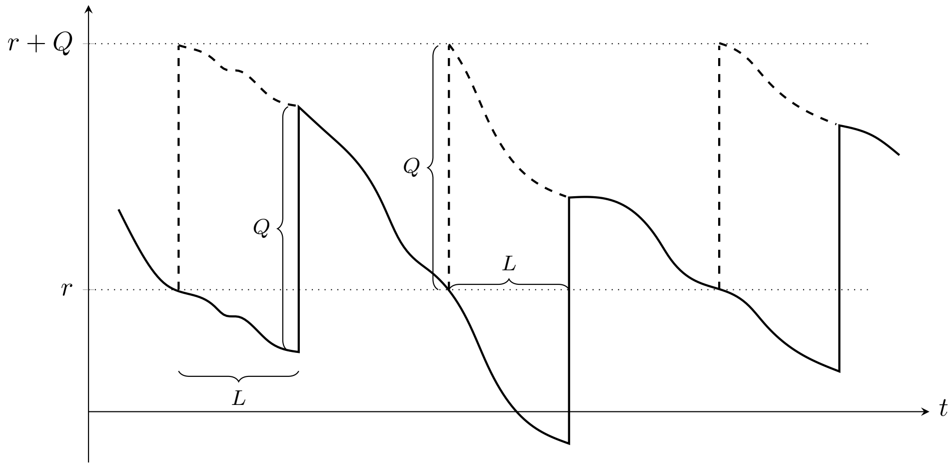


Figure 5.1 Inventory level (solid line) and inventory position (dashed line) under (r, Q) policy.

reorder point). Our goal is to determine the optimal r and Q to minimize the *expected cost* per year.

In a continuous-review setting, (r, Q) policies are equivalent to (s, S) policies (Section 4.4) as long as the inventory position equals s *exactly* at some point in every inventory cycle. This is guaranteed for continuous demand distributions (as in Sections 5.2–5.4) and for discrete demands in which each customer demands a single unit (as in Section 5.5). Recall that in an (s, S) policy, when the inventory position reaches s, we order up to S. Therefore, a given (r, Q) policy is equivalent to an (s, S) policy in which $s = r$ and $S = r + Q$. On the other hand, this equivalence does not hold for "lumpy" demand processes such as compound Poisson or for periodic-review systems, since in either case the inventory position may fall strictly below the reorder point before a replenishment order is placed.

In this chapter, we will focus first on the case in which the demands have a continuous distribution. We will discuss an exact model for this problem in Section 5.2, then discuss several common approximations in Section 5.3, and finally return to the exact model in Section 5.4 to prove some important properties of the optimal solution and its relationship to the economic order quantity with backorders (EOQB). Then, in Section 5.5, we discuss an exact model with discrete demands.

5.2 EXACT (r, Q) PROBLEM WITH CONTINUOUS DEMAND DISTRIBUTION

In this section, we introduce an exact model for systems with continuous demand distributions. We first formulate the expected cost function and then derive optimality conditions for it.

We continue to consider the usual costs: fixed cost $K \geq 0$, purchase cost $c \geq 0$, holding cost $h > 0$, and stockout cost $p > 0$. We'll use D to represent the lead-time demand; D is a random variable with mean μ, variance σ^2, pdf $f(d)$, and cdf $F(d)$. It is important to remember that D, μ, σ, etc. refer to lead-time demand, not to demand per year. Of course, the two are closely related. If the demand per year has mean λ and standard deviation τ and the lead time is L years, then the lead-time demand has mean λL and standard deviation $\tau\sqrt{L}$, assuming independence of demand across time.

5.2.1 Expected Cost Function

Our first step is to derive an exact expression for the expected cost as a function of r and Q. We place orders, on average, every Q/λ years (just as in the EOQ problem). Therefore, the expected fixed cost is given by $K\lambda/Q$. As in the EOQ, the annual purchase cost is given by $c\lambda$. Since it's independent of both Q and r, we'll ignore it in the cost calculations. It remains to evaluate the expected holding and stockout costs, which we will refer to collectively as the *inventory cost*. The inventory cost is incurred based on the inventory level, IL, a random variable whose distribution is difficult to determine for the same reasons as for periodic-review models with nonzero lead times; namely, that it depends on r and Q and that inventory decisions made at time t do not have an effect on IL until time $t + L$.

The solution to this problem is to use the conservation-of-flow concept discussed in Section 4.3.4.1, in which we relate the inventory level at time $t + L$ to the inventory position at time t (whose probability distribution, as we will see, is easy) and to the demand in the time interval $(t, t + L]$ (whose probability distribution we know). In particular, if the inventory position at time t is given by $IP(t)$, then the inventory level at time $t + L$ is given by

$$IL(t + L) = IP(t) - D(t, t + L], \tag{5.1}$$

where $D(t, t+L]$ is the cumulative demand that occurs between t and $t+L$. The reasoning is identical to that in Section 4.3.4.1, adjusted for continuous review: All of the items included in $IP(t)$—including items on hand and on order—will have arrived by time $t+L$, and no items ordered after time t will have arrived by time $t + L$. Therefore, all items that are on hand or on order at time t will be included in the inventory level at time $t+L$, except for the $D(t, t + L]$ items that have since been demanded.

As in the periodic-review case, we can drop the time indices from (5.1) in steady state and write

$$IL = IP - D, \tag{5.2}$$

where D is the lead-time demand. Zipkin (1986b) shows that (5.2) also holds—and therefore, so do many of the results in the rest of this section—under a range of stochastic lead-time settings.

Once we determine the distribution of IP, the (unconditional) expected inventory cost then follows from the law of total expectation. In particular, let $\bar{g}(x)$ be the rate at which the inventory cost accrues when $IL = x$:

$$\bar{g}(x) = hx^{+} + px^{-}. \tag{5.3}$$

($\bar{g}(\cdot)$ is a rate because the inventory level is changing continuously over time, given in units of money per year.) Then the expected inventory cost per year is

$$\begin{aligned}
\mathbb{E}[\text{inventory cost}] &= \mathbb{E}_{IL}\left[\bar{g}(IL)\right] \\
&= \mathbb{E}_{IP}\left[\mathbb{E}_{IL|IP}\left[\bar{g}(IL)\right]\right] \\
&= \mathbb{E}_{IP}\left[\mathbb{E}_{D}\left[\bar{g}(IP - D)\right]\right] \\
&= \mathbb{E}_{IP}[g(IP)],
\end{aligned} \tag{5.4}$$

where

$$g(y) = h\mathbb{E}[(y-D)^+] + p\mathbb{E}[(D-y)^+] \tag{5.5}$$

is the rate at which the expected inventory cost accrues at time $t+L$ when the inventory position at time t equals y. The expectation in (5.5) is over the lead-time demand. Note that $g(r,Q)$, with two arguments, is the expected total expected cost, whereas $g(y)$, with one argument, is the expected inventory cost.

$g(y)$ is simply the newsvendor expected cost function (Section 4.3.2). Let S^* be its optimizer, given by (4.17).

It remains to determine the distribution of IP. By the definition of an (r,Q) policy, we know that IP takes values only in $[r, r+Q]$. It turns out that IP has a very simple distribution—it is uniform on $[r, r+Q]$, under some mild conditions on the lead-time demand distribution (Serfozo and Stidham 1978, Browne and Zipkin 1991). Therefore, (5.4) implies that

$$\mathbb{E}[\text{inventory cost}] = \frac{1}{Q}\int_r^{r+Q} g(y)dy. \tag{5.6}$$

Combining the expected inventory cost (5.6) and the expected fixed cost $K\lambda/Q$, we get the following expression for the expected total cost per year:

$$g(r,Q) = \frac{K\lambda + \int_r^{r+Q} g(y)dy}{Q}. \tag{5.7}$$

For early derivations of this equation, see, e.g., Hadley and Whitin (1963).

Zheng (1992) proves the following:

Lemma 5.1 *$g(r,Q)$ is jointly convex in r and Q.*

Proof. Let

$$I(r,Q) = \frac{1}{Q}\int_r^{r+Q} \mathbb{E}[(y-D)^+]dy$$
$$B(r,Q) = \frac{1}{Q}\int_r^{r+Q} \mathbb{E}[(D-y)^+]dy$$

be the expected on-hand inventory and backorders, respectively, as functions of r and Q. Then we can write

$$g(r,Q) = \frac{K\lambda}{Q} + hI(r,Q) + pB(r,Q).$$

Moreover,

$$\begin{aligned}
I(r,Q) &= \frac{1}{Q}\int_r^{r+Q} \mathbb{E}[(y-D) + (y-D)^-]dy \\
&= \frac{1}{Q}\int_r^{r+Q} [y - \mathbb{E}[D] + \mathbb{E}[(D-y)^+]]dy \\
&= \frac{1}{Q}\left(\frac{y^2}{2} - \lambda L y\right|_r^{r+Q} + B(r,Q) \\
&= \frac{Q}{2} + r - \lambda L + B(r,Q),
\end{aligned}$$

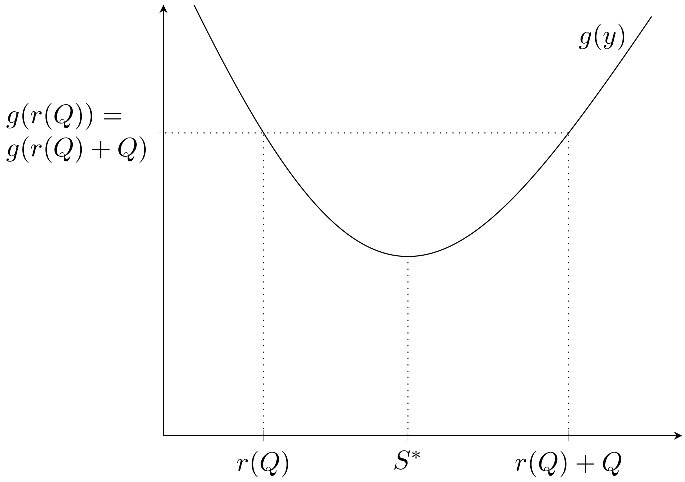


Figure 5.2 Inventory costs are equal at start and end of replenishment cycle.

so

$$g(r, Q) = \frac{K\lambda}{Q} + h\left(\frac{Q}{2} + r - \lambda L\right) + (h + p)B(r, Q). \tag{5.8}$$

The first and second terms are clearly convex; the joint convexity of $B(r, Q)$ is proven by Zipkin (1986a). ■

In what follows, we use the expected cost expression (5.7) to derive optimality conditions for r and Q by first fixing Q and finding the optimal corresponding r, and then optimizing over Q. Although these conditions tell us when a given solution is optimal, they do not give us an algorithm for finding such solutions. Before developing such an algorithm, we first discuss several common approximations for finding the optimal parameters for an (r, Q) policy, in Section 5.3. We then return to the exact model in Section 5.4, proving properties of these optimal solutions that we can use to develop an algorithm.

5.2.2 Optimality Conditions

We will optimize sequentially: $\min_Q \{\min_r g(r, Q)\}$. Let $r(Q)$ be the optimal r for a given Q.

Lemma 5.2 *For any $Q > 0$, $r = r(Q)$ if and only if*

$$g(r) = g(r + Q). \tag{5.9}$$

Proof. Follows immediately from the first-order condition:

$$\frac{\partial g(r, Q)}{\partial r} = \frac{g(r + Q) - g(r)}{Q} = 0.$$

■

The inventory position equals $r + Q$ at the start of a replenishment cycle (just after an order is placed) and equals r at the end (just before the next order is placed). Therefore, Lemma 5.2 says that, for a given Q, the optimal r makes the inventory cost rates equal at the start and end of the replenishment cycle. (See Figure 5.2.) In between, the inventory costs are lower, due to the convexity of $g(y)$.

The motivation behind this result is that, during one replenishment cycle, we need to pass through all of the inventory positions in $[r, r+Q]$, and we spend an equal amount of time in each. For fixed Q, we minimize the total cost by choosing the r that keeps $g(y)$ as small as possible over those inventory positions. Since $g(y)$ is convex, the r that keeps $g(y)$ as small as possible over $[r, r+Q]$ is the r for which $g(r) = g(r+Q)$.

This result can be visualized as follows. Imagine a two-dimensional bowl shaped like the function $g(y)$. For a given Q, we can find the optimal value of r by dropping a horizontal bar of length Q into the bowl; then $r(Q)$ equals the height of the bar when it comes to rest.

We can now characterize the optimal (r, Q) pair.

Theorem 5.3 (r, Q) *minimize* $g(r, Q)$ *if and only if*

$$g(r, Q) = g(r+Q) = g(r). \tag{5.10}$$

Proof. From (5.7),

$$\begin{aligned}\frac{\partial g}{\partial Q} &= \frac{Qg(r+Q) - \left[K\lambda + \int_r^{r+Q} g(y)dy\right]}{Q^2} \\ &= \frac{g(r+Q) - g(r,Q)}{Q} = 0\end{aligned}$$

This proves the first equality. The second follows from Lemma 5.2. ■

☐ **EXAMPLE 5.1**

Recall Joe's Corner Store from Example 3.1. Suppose now that the annual demand for candy bars is normally distributed with a mean of 1300 and a standard deviation of 150. Joe's customers are fiercely loyal, both to Joe and to his brand of candy, so if the store is out of stock, they are willing to wait for their candy. (That is, demands are backordered, not lost.) However, each stockout costs \$0.50 in lost profit and \$7.00 in loss of goodwill per year. The lead time is $L = 1/12$ year. What are the optimal r and Q?

We have $K = 8$, $h = 0.225$, and $p = 7.5$. The lead-time demand has parameters $\mu = 1300/12 = 108.3$ and $\sigma = 150/\sqrt{12} = 43.3$.

Let $Q = 328.5$. Then $r(Q) = 126.8$ by Lemma 5.2 since

$$g(126.8) = g(126.8 + 328.5) = 78.1.$$

From (5.7),

$$g(126.8, 328.5) = 78.1,$$

confirming via Theorem 5.3 that $(r, Q) = (126.8, 328.5)$ is optimal for this instance. ☐

Theorem 5.3 says that, surprisingly, not only are the *inventory* costs equal at the start and end of the replenishment cycle, but these costs are also equal to the *total* cost per year. For some very simple demand distributions, the simultaneous equations (5.10) can be solved analytically. More commonly, though, (5.10) must be solved using an iterative algorithm. In order to derive such an algorithm, we will need some additional properties of the model. Before delving into those, however, we will shift our attention to approximate models.

5.3 APPROXIMATIONS FOR (r, Q) PROBLEM WITH CONTINUOUS DISTRIBUTION

5.3.1 Expected-Inventory-Level Approximation

The first approximation we discuss is probably the best known and most widely covered approximation to find r and Q. (Unfortunately, it is also one of the least accurate; see Section 5.3.5.) It dates back to Whitin (1953) (whose book in fact contains one of the earliest attempts to optimize r and Q simultaneously) as well as to subsequent developments by Hadley and Whitin (1963). We call this the *expected-inventory-level (EIL) approximation*, for reasons that will become clear shortly.

The approach relies on the following two simplifying assumptions to make the model tractable:

- *Simplifying Assumption 1* (SA1): We incur holding costs at a rate of $h \cdot IL$ per year, where IL is the inventory level, *whether IL is positive or negative*.
- *Simplifying Assumption 2* (SA2): The stockout cost is charged once per unit of unmet demand, not per year.

Neither assumption is particularly realistic, but we make them for mathematical convenience. SA1 is obviously untrue, since it suggests we *earn* a holding "credit" when $IL < 0$, but it is not too inaccurate if the expected number of stockouts is small. SA2 is not as outrageous, but it is not typical, either in practice or in other inventory models. (Actually, SA1 would not be problematic at all if we didn't also assume SA2. If the stockout cost were charged per year, then we could simply replace the stockout cost p with $p + h$, thus canceling the artificial "credit" of h for negative inventory.)

5.3.1.1 Expected Cost Function

In this section, we will derive an expression for the approximate expected cost per year as a function of the decision variables Q and r.

Holding Cost: Figure 5.3 contains a graph of the *expected* inventory over time. s is the expected on-hand inventory when the order arrives:

$$s = r - \lambda L.$$

In other words, s is the safety stock—the extra inventory held on hand to meet demand in excess of the mean.

The average inventory level is

$$s + \frac{Q}{2} = r - \lambda L + \frac{Q}{2}. \tag{5.11}$$

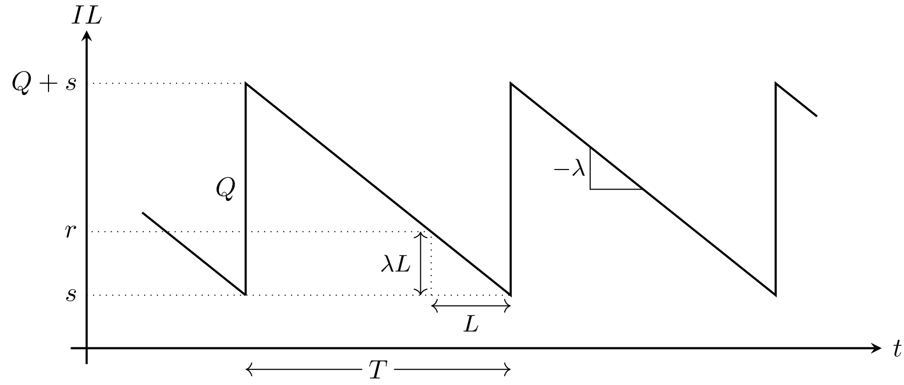


Figure 5.3 Expected inventory curve for (r, Q) policy.

By SA1, the expected holding cost per year is

$$h\left(r - \lambda L + \frac{Q}{2}\right). \tag{5.12}$$

Of course, this expression is only approximate. The essence of the approximation is that we are calculating the expected holding cost as $h \cdot \mathbb{E}[IL] = h \cdot \mathbb{E}[IL]^+$ (provided that $\mathbb{E}[IL] > 0$), whereas it actually equals $h \cdot \mathbb{E}[IL^+]$, and the two are not equal. That is why we refer to this as the "expected-inventory-level" approximation. The problem is more difficult without SA1 because of the nonlinearity introduced by the $[\cdot]^+$ operator. As previously noted, the EIL approximation becomes less accurate as the expected number of stockouts increases or, equivalently, as s decreases.

Fixed Cost: The expected fixed cost per year is given by K times the expected number of orders per year. From Figure 5.3, we see that $\mathbb{E}[T] = Q/\lambda$. Therefore, the expected cost per year is

$$\frac{K\lambda}{Q}. \tag{5.13}$$

Stockout Cost: The expected number of stockouts per order cycle is given by

$$\mathbb{E}[(D - r)^+] = \int_r^\infty (d - r) f(d) dd = n(r), \tag{5.14}$$

where $n(r)$ is the *loss function* for the lead-time demand distribution. (See Section 4.3.2.2 or Section C.3.1.) The expected number of stockouts per year is $n(r)/\mathbb{E}[T] = \lambda n(r)/Q$. By SA2, the expected stockout cost per year is simply

$$\frac{p\lambda n(r)}{Q}. \tag{5.15}$$

Note that we are assuming that $r > 0$, which is a reasonable assumption in practice. (The reason we make simplifying assumption SA2 is that if the stockout cost were charged per year, then the integrand in the expected stockout cost per year would contain $(d - r)^2$ in place of $(d - r)$, and this would be significantly harder to analyze. See Problem 5.23.)

Total Cost: Combining (5.12), (5.13), and (5.15), we get the total expected cost per year:

$$g(r, Q) = h\left(r - \lambda L + \frac{Q}{2}\right) + \frac{K\lambda}{Q} + \frac{p\lambda n(r)}{Q}. \tag{5.16}$$

5.3.1.2 Solution As in the EOQ model, we will optimize by setting the first derivative to 0. Since there are two decision variables, we must take partial derivatives with respect to each and set them both to 0:

$$\begin{aligned} \frac{\partial g}{\partial Q} &= \frac{h}{2} - \frac{K\lambda}{Q^2} - \frac{p\lambda n(r)}{Q^2} = 0 \\ \iff \frac{1}{Q^2}[K\lambda + p\lambda n(r)] &= \frac{h}{2} \\ \iff Q^2 &= \frac{2[K\lambda + p\lambda n(r)]}{h} \end{aligned}$$

or

$$Q = \sqrt{\frac{2\lambda[K + pn(r)]}{h}}. \tag{5.17}$$

And:

$$\begin{aligned} \frac{\partial g}{\partial r} &= h + \frac{p\lambda n'(r)}{Q} = 0 \\ \iff h + \frac{p\lambda(F(r) - 1)}{Q} &= 0 \end{aligned}$$

(using (C.15)), so

$$r = F^{-1}\left(1 - \frac{Qh}{p\lambda}\right). \tag{5.18}$$

Now we have two equations with two unknowns, but these equations cannot be solved in closed form. The approach given in Algorithm 5.1 first sets Q equal to the EOQ quantity, i.e., ignoring the demand randomness. It then proceeds iteratively, solving (5.18) to find r, solving (5.17) to find Q, and so on. The algorithm terminates when one (or both) of the parameters haven't changed much since the last iteration. (ϵ is the convergence tolerance.) Hadley and Whitin (1963) prove that this algorithm converges to the optimal r and Q for (5.16)—though it's important to keep in mind that (5.16) itself is only an approximate cost function.

Typically, $Q < \lambda$ and $h < p$, so that the argument to F^{-1} in (5.18) is between 0 and 1. In rarer cases, however, Qh may be larger than $p\lambda$, in which case the argument to F^{-1} is negative and there is no solution to (5.18). If this happens, we can simply set r to its minimum allowable value (which we have assumed is 0).

☐ **EXAMPLE 5.2**

Let us apply the EIL approximation to Joe's Corner Store from Example 5.1. Using Algorithm 5.1, we first set Q equal to the EOQ quantity, which we know from Example 3.1 to be 304.1. From (5.18), we have

$$r = F^{-1}\left(1 - \frac{304.1 \cdot 0.225}{7.5 \cdot 1300}\right) = F^{-1}(0.9930) = 214.7.$$

Algorithm 5.1 Iterative algorithm for EIL approximation for (r, Q) policy

```
1: Q ← √(2Kλ/h)                                        ▷ Initialization: use EOQ
2: r ← ∞
3: repeat                                              ▷ Main loop
4:     Q_prev ← Q; r_prev ← r                          ▷ Remember previous values
5:     r ← r that solves (5.18), or 0 if none          ▷ Solve for r using current Q
6:     Q ← Q that solves (5.17)                        ▷ Solve for Q using current r
7: until |Q − Q_prev| ≤ ϵ and/or |r − r_prev| ≤ ϵ      ▷ Termination check
8: return (r, Q)
```

Now, to calculate Q, we'll need to calculate $n(r)$. We can calculate $n(r)$ using $\mathscr{L}(z)$, the standard normal loss function, via (C.31), where $z = (r - \mu)/\sigma$. $\mathscr{L}(z)$, in turn, can be calculated using (C.22).

If $r = 214.7$, then $z = (r - \mu)/\sigma = 2.456$, $\mathscr{L}(z) = 0.002292$, and $n(r) = 0.002292 \cdot 43.3 = 0.0993$. Then, from (5.17), we have

$$Q = \sqrt{\frac{2 \cdot 1300[8 + 7.5 \cdot 0.0993]}{0.225}} = 317.9.$$

Repeating this process:

$$\begin{aligned} r &= F^{-1}\left(1 - \frac{317.9 \cdot 0.225}{7.5 \cdot 1300}\right) = F^{-1}(0.9927) = 214.0 \\ \implies n(r) &= 0.1042 \\ Q &= \sqrt{\frac{2 \cdot 1300[8 + 7.5 \cdot 0.1042]}{0.225}} = 318.6 \\ r &= F^{-1}\left(1 - \frac{318.6 \cdot 0.225}{7.5 \cdot 1300}\right) = F^{-1}(0.9927) = 214.0 \end{aligned}$$

Because r did not change since the previous iteration, the process can terminate. We set $r = 214.0$, $Q = 318.6$. The *approximate* annual expected cost of this solution, using (5.16), is

$$\begin{aligned} g(r, Q) =& 0.225\left(214.0 - \frac{1300}{12} + \frac{318.6}{2}\right) + \frac{8 \cdot 1300}{318.6} \\ &+ \frac{7.5 \cdot 1300 \cdot 0.1042}{318.6} \\ =& 95.45. \end{aligned}$$

The *exact* expected cost, using (5.7), is 92.29, 18.2% larger than the optimal cost of 78.1 from Example 5.1. □

5.3.1.3 Service Levels One major limitation of (r, Q) policies as formulated above is that p is very hard to estimate. But there is a close relationship between p and the service level (see Section 4.3.4.2): As p increases, it's more costly to stock out, so the service level should increase. In practice, many firms would rather omit the stockout cost from the objective function and add a constraint requiring the service level to be at least a certain value.

First suppose that we wish to impose a type-1 service level constraint. That is, we want to require the probability that no stockouts occur in a given cycle to be at least α. Since stockouts occur if and only if the lead-time demand is greater than r, this probability is simply $F(r)$. The expected cost function we wish to minimize is identical to (5.16) except it no longer contains a term for the stockout cost. Therefore, we need to solve

$$\text{minimize} \qquad g(r, Q) = h\left(r - \lambda L + \frac{Q}{2}\right) + \frac{K\lambda}{Q} \tag{5.19}$$

$$\text{subject to} \qquad F(r) \geq \alpha \tag{5.20}$$

At optimality, the constraint (5.20) will always hold as an equality. (Why?) Therefore, the optimal reorder point is given by $r = F^{-1}(\alpha)$. If the lead-time demand is normally distributed, then the optimal reorder point is

$$r = \mu + z_\alpha \sigma. \tag{5.21}$$

As we know from Section 4.3.2, this is exactly the form of the optimal solution to the newsvendor problem. As in the newsvendor problem, the first term of (5.21) represents the cycle stock (to meet the expected demand during the lead time), while the second term represents the safety stock (to meet excess demand during the lead time), since the safety stock is given by $s = r - \mu$.

What about Q? Well, once r is fixed, we can ignore the constraint, and the term $h(r - \lambda L)$ in the objective function (5.19) is a constant. What's left in (5.19) is exactly equal to the EOQ cost function (3.3). Therefore, we set Q to the EOQ value.

The expected cost of this solution is given by

$$\begin{aligned} g(r, Q) &= hz_\alpha \sigma + \frac{hQ}{2} + \frac{K\lambda}{Q} \\ &= hz_\alpha \sigma + \sqrt{2K\lambda h}. \end{aligned} \tag{5.22}$$

(The first equality follows from the fact that μ, the mean lead-time demand, equals λL. The second equality follows from (3.5).) This is an *exact* solution to the *approximate* model with a type-1 service level constraint. This approach is often used as an approximation even when p is known; see Section 5.3.3. It is important in other ways, as well; for example, we will make use of it when we discuss the location model with risk pooling (LMRP) in Section 12.2.

Now consider a type-2 service level constraint; we want to require the fill rate to be at least β. We know that the average proportion of demands that stock out in each cycle is $n(r)/Q$, so we need to replace (5.20) with

$$\frac{n(r)}{Q} = 1 - \beta. \tag{5.23}$$

The resulting problem is significantly harder to solve: Since (5.23) contains both Q and r, we can no longer solve first for r and then solve independently for Q. Nevertheless, a reasonable approximation is simply to set $Q =$ EOQ (as in the case of type-1) and compute r using $n(r) = Q(1 - \beta)$. There is a more accurate method that involves a more complex formula for Q that is solved simultaneously with (5.18); see Nahmias (2005) for details.

☐ **EXAMPLE 5.3**

Return to Example 5.2 and suppose that Joe wishes to ensure a type-1 service level of $\alpha = 0.98$. What are the optimal r and Q? What about for a type-2 service level of $\beta = 0.98$?

For the type-1 service level constraint, we have $z_\alpha = \Phi^{-1}(0.98) = 2.0538$ and

$$\begin{aligned} r &= 108.3 + 2.0538 \cdot 43.3 = 197.3 \\ Q &= \text{EOQ} = 304.1 \end{aligned}$$

Using the approximate approach for the type-2 constraint, we have $Q = \text{EOQ} = 304.1$. We need to solve

$$n(r) = 304.1(0.02) = 6.081.$$

You can confirm that this equation is satisfied by $r = 139.1$. ☐

5.3.2 EOQB Approximation

There are important connections between the EOQ problem with planned backorders (EOQB; Section 3.5) and (r, Q) policies with continuous demand distributions. We explore these connections further in Section 5.4. The *EOQB approximation* for finding near-optimal r and Q makes use of the EOQB, setting Q using (3.27) and r using Lemma 5.2. This approach has a fixed worst-case error bound of $\frac{1}{8}$ that we will prove in Section 5.4, and an even tighter bound of 11.8% (which we will not prove).

☐ **EXAMPLE 5.4**

If we use the EOQB approximation to solve the problem in Example 5.2, we get

$$Q = \sqrt{\frac{2 \cdot 8 \cdot 1300(0.225 + 7.5)}{0.225 \cdot 7.5}} = 308.6.$$

You can confirm that $r = 128.6$ solves

$$g(r) = g(r + 308.6).$$

The solution $(r, Q) = (128.6, 308.6)$ has an expected annual cost of 78.2, only 0.26% larger than the optimal cost from Example 5.1 and much less than the worst-case bound of 11.8%. It is also considerably better than the solution from the EIL approximation in Example 5.2. ☐

5.3.3 EOQ+SS Approximation

Another common approximation for r and Q is to convert the inventory-cost parameters into a service level and then to use the approach described in Section 5.3.1.3 for type-1 service level constraints. In particular,

$$Q = \sqrt{\frac{2K\lambda}{h}}$$

$$r = \mu + z_\alpha \sigma,$$

where $\alpha = p/(p + h)$. The safety stock is given by $s = r - \mu = z_\alpha \sigma$. The expected inventory process can be thought of as being decomposed into two parts, a "top" part that looks like an EOQ curve and a "bottom" part that is flat, with a height of s, the safety stock. We therefore refer to this as the *EOQ+SS approximation*.

The EOQ+SS approximation should not be confused with the EOQB approximation discussed in Section 5.3.2. Although both approaches use the EOQ(B) model to approximate an (r, Q) policy, they do so in different ways. Importantly, the EOQ+SS approximation does *not* have a fixed worst-case error bound (see Problem 5.18), although some authors mistakenly apply Zheng's (1992) worst-case bound of $\frac{1}{8}$ to it. Nevertheless, it is a reasonable approximation that performs well if $\alpha = p/(p + h)$ provides an acceptable service level.

☐ **EXAMPLE 5.5**

The EOQ+SS approximation yields the following solution for the problem in Example 5.2:

$$Q = \sqrt{\frac{2 \cdot 8 \cdot 1300}{0.225}} = 304.1$$
$$r = 108.3 + 1.8938 \cdot 43.3 = 190.3$$

since $\alpha = 0.9709$ and $z_\alpha = 1.8938$. The solution $(r, Q) = (190.3, 304.1)$ has an expected annual cost of 87.1, or 11.5% worse than optimal. ☐

A similar approach can be used when the lead time itself is stochastic. Suppose the lead time L has mean μ_L and standard deviation σ_L (in years). Then the lead-time demand has mean and variance

$$\mu = \lambda \mu_L \tag{5.24}$$
$$\sigma^2 = \lambda^2 \sigma_L^2 + \mu_L \tau^2, \tag{5.25}$$

where, as usual, λ and τ^2 are the mean and variance of the demand per year. (See Problem 5.16.) Equations (5.21) and (5.22) still hold under these new definitions of μ and σ. This approach is used in Case Study 5.1.

5.3.4 Loss-Function Approximation

From (5.8),

$$g(r, Q) = \frac{K\lambda}{Q} + h\left(\frac{Q}{2} + r - \lambda L\right) + (h + p)B(r, Q),$$

where

$$B(r, Q) = \frac{1}{Q}\int_r^{r+Q} \mathbb{E}[(D - y)^+]dy = \frac{1}{Q}\int_r^{r+Q} n(y)dy$$

by (C.12). Let $n^{(2)}(x)$ be the *second-order loss function* for the lead-time demand distribution (see Section C.3.1):

$$n^{(2)}(x) = \frac{1}{2}\mathbb{E}\left[\left([X - x]^+\right)^2\right] = \int_x^\infty n(y)dy. \tag{5.26}$$

Then we can rewrite $B(r, Q)$ as

$$B(r, Q) = \frac{1}{Q}\left[n^{(2)}(r) - n^{(2)}(r+Q)\right].$$

Therefore,

$$g(r, Q) = \frac{K\lambda}{Q} + h\left(\frac{Q}{2} + r - \lambda L\right) + \frac{h+p}{Q}\left[n^{(2)}(r) - n^{(2)}(r+Q)\right]. \tag{5.27}$$

Let's consider the $n^{(2)}(r+Q)$ term. We typically set r so that stockouts are unlikely during the lead time, i.e., so that the lead-time demand is unlikely to exceed r. It is therefore even less likely to exceed $r+Q$. Since $n^{(2)}(r+Q)$ equals the expected value of the square of the amount by which the lead-time demand exceeds $r+Q$, it, too, is likely to be small. For example, using the parameters in Example 5.2 and $(r, Q) = (126.8, 328.5)$ from Example 5.1, $n^{(2)}(r+Q)$ is less than 10^{-13}.

Therefore, Hadley and Whitin (1963) propose assuming $n^{(2)}(r+Q) \approx 0$ and then approximating $g(r, Q)$ as

$$g(r, Q) = \frac{K\lambda}{Q} + h\left(\frac{Q}{2} + r - \lambda L\right) + \frac{h+p}{Q}n^{(2)}(r).$$

Taking partial derivatives, we get

$$\frac{\partial g}{\partial Q} = -\frac{K\lambda}{Q^2} + \frac{h}{2} - \frac{(h+p)n^{(2)}(r)}{Q^2} = 0$$

$$Q = \sqrt{\frac{2\left[K\lambda + (h+p)n^{(2)}(r)\right]}{h}} \tag{5.28}$$

and

$$\frac{\partial g}{\partial r} = h - \frac{(h+p)n(r)}{Q} = 0$$

$$\Longrightarrow\ n(r) = \frac{hQ}{h+p} \tag{5.29}$$

using the fact that $\frac{d}{dx}n^{(2)}(x) = -n(x)$ (see (C.20)). Equations (5.28) and (5.29) can be solved for r and Q using an iterative method similar to that for the EIL approximation in Algorithm 5.1.

In fact, a similar approach can be used directly on (5.27), iteratively solving two optimality equations analogous to (5.28) and (5.29). This approach provides an exact (not heuristic) solution to find the optimal parameters for an (r, Q) policy (Farvid and Rosling 2014).

☐ **EXAMPLE 5.6**

We will use the loss-function approximation for the problem in Example 5.2. We first set Q equal to the EOQ quantity, 304.1. Setting $r = 129.1$ satisfies (5.29) since

$$n(129.1) = 8.8557 = \frac{0.225 \cdot 304.1}{0.225 + 7.5}.$$

Then $n^{(2)}(129.1) = 204.6487$. Therefore, from (5.28),

$$Q = \sqrt{\frac{2\left[8 \cdot 1300 + (0.225 + 7.5) \cdot 204.6487\right]}{0.225}} = 326.3.$$

Repeating this process:

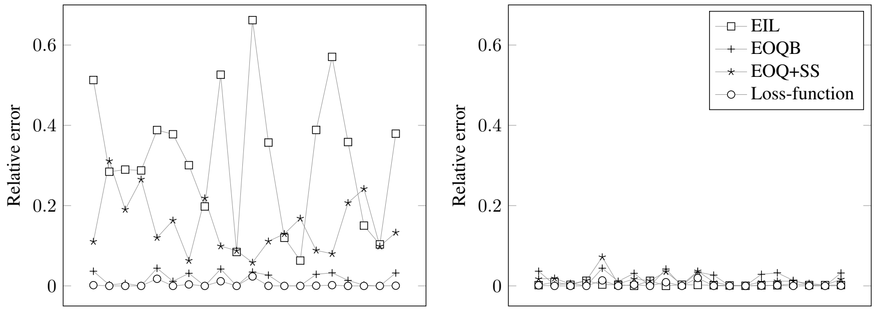
$$\begin{aligned}
n(127.1) = 9.5050 &= \frac{0.225 \cdot 326.3}{0.225 + 7.5} \\
\Longrightarrow\ r &= 127.1,\ n^{(2)}(r) = 223.0154 \\
Q &= \sqrt{\frac{2\left[8 \cdot 1300 + (0.225 + 7.5) \cdot 223.0154\right]}{0.225}} = 328.3 \\
n(126.9) = 9.5611 &= \frac{0.225 \cdot 328.3}{0.225 + 7.5} \\
\Longrightarrow\ r &= 126.9,\ n^{(2)}(r) = 224.6196 \\
Q &= \sqrt{\frac{2\left[8 \cdot 1300 + (0.225 + 7.5) \cdot 224.6196\right]}{0.225}} = 328.4 \\
n(126.9) = 9.5611 &= \frac{0.225 \cdot 328.4}{0.225 + 7.5} \\
\Longrightarrow\ r &= 126.9
\end{aligned}$$

Because r did not change since the previous iteration, the process can terminate. We have $r = 126.9$, $Q = 328.4$. This is the optimal solution (within rounding error), as found in Example 5.1. □

5.3.5 Performance of Approximations

Figure 5.4(a) plots the relative error of each of the four approximations described above on 20 randomly generated instances. The relative error is calculated as $(g(r, Q) - g(r^*, Q^*))/g(r^*, Q^*)$, where (r, Q) is the solution returned by the approximation, (r^*, Q^*) is the optimal solution, and $g(\cdot, \cdot)$ is the exact cost function, given by (5.7). The mean and maximum relative error are given in the first set of columns in Table 5.1. Despite the fact that they are perhaps the two most commonly taught and used approaches, the EIL and EOQ+SS approximations perform the worst, with mean relative errors of over 30% and 14%, respectively. The other two approximations perform much better, with mean errors below 2%. On the other hand, they are more difficult to implement, since they require solving (5.9) (in the EOQB approximation) or computing $n^{(2)}(\cdot)$ (in the loss-function approximation).

In Theorem 5.7, we will show that the (r, Q) cost is relatively insensitive to errors in Q. This suggests that the poor performance of the EIL and EOQ+SS is largely driven by their poor choices of r, rather than of Q. Indeed, if we alter each of the approximations to discard r at the end and instead set $r = r(Q)$, the performance is much better, with mean errors below 2% for all four approximations; see Figure 5.4(b) and the second set of columns in Table 5.1. (Note that the performance of the EOQB approximation is the same in both experiments, since that approximation already sets $r = r(Q)$.)



(a) Original approximations. (b) Approximations with r set to $r(Q)$.

Figure 5.4 Relative error of (r, Q) approximations.

Table 5.1 Mean and maximum error of (r, Q) approximations.

	Original		With $r(Q)$	
Approximation	Mean	Max	Mean	Max
EIL	0.320	0.662	0.003	0.013
EOQB	0.017	0.044	0.017	0.044
EOQ+SS	0.147	0.311	0.015	0.072
Loss-function	0.003	0.024	0.002	0.020

5.4 EXACT (r, Q) PROBLEM WITH CONTINUOUS DISTRIBUTION: PROPERTIES OF OPTIMAL r AND Q

We now return to the exact model from Section 5.2. We have two main goals in this section. First, we will analyze the properties of optimal solutions (and their costs) for (r, Q) policies, by deriving optimality conditions for r and Q and then proving properties of the resulting optimal solutions. Second, we will compare (r, Q) policies to the EOQB model and prove that, if the EOQB model is used as a heuristic for optimizing r and Q, as discussed in Section 5.3.2, the resulting error has a fixed bound. We do this by treating the EOQB as a deterministic (r, Q) policy, a reasonable interpretation since the two models include the same costs and both allow backorders. Our analysis in this section is based primarily on the work of Zheng (1992).

Let $G(Q)$ equal the expected cost per year as a function of Q, assuming r is set optimally for that Q—that is,

$$G(Q) = g(r(Q), Q). \tag{5.30}$$

Let $H(Q)$ be the value of $g(y)$ at $y = r(Q)$ or, equivalently, at $r(Q) + Q$:

$$H(Q) = g(r(Q)) = g(r(Q) + Q). \tag{5.31}$$

One can show (see Problem 5.8) that

$$\int_{r(Q)}^{r(Q)+Q} g(y)dy = \int_0^Q H(y)dy. \tag{5.32}$$

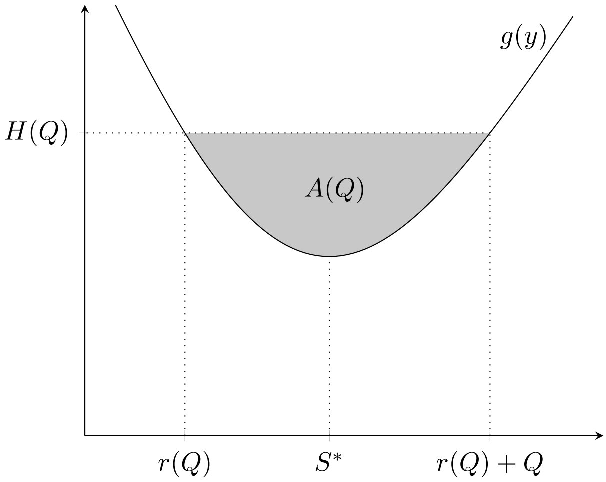


Figure 5.5 $A(Q)$ and $H(Q)$.

Therefore, from (5.7), we can write

$$G(Q) = \frac{K\lambda + \int_0^Q H(y)dy}{Q}, \tag{5.33}$$

which expresses the expected total cost as a function of Q only, not r. One can show that $G(Q)$ is convex. Finally, let

$$A(Q) = QH(Q) - \int_0^Q H(y)dy \tag{5.34}$$

be the area between $g(y)$ and the line at height $H(Q)$; see Figure 5.5.

The following theorem provides a surprisingly simple condition under which Q minimizes $G(Q)$ (and therefore $(r(Q), Q)$ minimizes $g(r, Q)$). We'll use Q^* to denote the minimizer of $G(Q)$.

Theorem 5.4 *Q minimizes $G(Q)$, i.e., $Q = Q^*$, if and only if*

$$A(Q) = K\lambda. \tag{5.35}$$

Proof. From (5.34),

$$\begin{aligned} A(Q) &= Qg(r(Q)+Q) - \int_0^Q H(y)dy \\ &= Qg(r(Q)+Q) - [Qg(r(Q), Q) - K\lambda] \end{aligned}$$

by (5.30) and (5.33). At optimality, this equals

$$Qg(r(Q)+Q) - [Qg(r(Q)+Q) - K\lambda] = K\lambda$$

by Theorem 5.3. ■

Therefore, the optimal length of the bar to drop into the $g(y)$ "bowl" is the Q such that the area between the bar and the bowl equals $K\lambda$. Unfortunately, we can't generally determine Q^* in closed form, since $A(Q)$ depends on $H(Q)$, which in turn depends on $r(Q)$, which also cannot be found in closed form. However, Q^* can be found through a straightforward search; see Section 5.4.1.

☐ **EXAMPLE 5.7**

Recall from Example 5.1 that the optimal parameters for Joe's Corner Store are $r = 126.8$, $Q = 328.5$. We already know that $g(126.8) = g(126.8 + 328.5) = 78.1$, which means that $H(Q)$ also equals 78.1. Via numerical integration, we have

$$\int_0^Q H(Q) = 15{,}246.2,$$

so

$$A(Q) = 328.5 \cdot 78.1 - 15{,}246.2 = 10{,}410,$$

which equals $K\lambda$ within rounding error. (More digits of precision in Q and $r(Q)$ would result near-exact equality.) This provides an alternate confirmation, via Theorem 5.4, that $(r, Q) = (126.8, 328.5)$ are the optimal parameters. ☐

5.4.1 Optimization of r and Q

Algorithm 5.2 uses Theorem 5.4 to find the exact optimal values of r and Q for a continuous-review (r, Q) policy with continuously distributed demand. The algorithm is basically a bisection search over Q, with an inner step that finds $r(Q)$ for each candidate value of Q. The bounds in the initialization step come from Theorem 5.5, below. In the termination criterion, ϵ is the desired tolerance.

Algorithm 5.2 Exact algorithm for continuous-review (r, Q) policy with continuous demand distribution

$\underline{Q} \leftarrow Q_d^*$; $\overline{Q} \leftarrow Q_0$ from Theorem 5.5 ▷ Initialization
repeat ▷ Main loop
  $Q \leftarrow (\underline{Q} + \overline{Q})/2$ ▷ Candidate value for Q
  $r \leftarrow r(Q)$, where $r(Q)$ satisfies (5.9) ▷ Optimal r for Q
  $A \leftarrow A(Q)$ ▷ $A(Q)$
  if $A > K\lambda$ **then** $\overline{Q} \leftarrow Q$ ▷ Update bounds on Q
  else if $A < K\lambda$ **then** $\underline{Q} \leftarrow Q$
  end if
until $|A - K\lambda| \le \epsilon$ ▷ Termination check via Theorem 5.4
return (r, Q)

5.4.2 Noncontrollable and Controllable Costs

Recall that S^* is the minimizer of $g(y)$. Let

$$H_0(Q) = H(Q) - g(S^*). \tag{5.36}$$

Then we can rewrite the cost function as

$$G(Q) = g(S^*) + G_0(Q), \tag{5.37}$$

where

$$G_0(Q) = \frac{K\lambda + \int_0^Q H_0(y)dy}{Q}.$$

The first term in (5.37), $g(S^*)$, represents the *noncontrollable cost* in the (r, Q) policy. Even if we could keep the inventory position at S^* at all times, by constantly placing orders, we could not avoid the cost $g(S^*)$—it is a consequence of the randomness in the demand. Of course, we cannot constantly place orders (since there is a fixed cost for each order), so the inventory position will deviate from the ideal level S^*, and the inventory costs will increase from $g(S^*)$. By varying the order quantity Q, we adjust the trade-off between fixed and inventory costs. The increase in cost over and above $g(S^*)$ is the *controllable cost*, and this is captured by $G_0(Q)$, the second term of (5.37).

5.4.3 Relationship to EOQB

As we know from Section 5.3.2, the EOQB (Section 3.5) provides an approximation of an (r, Q) policy. In fact, we can view the EOQB as a special case of an (r, Q) policy obtained by assuming the lead-time demand is deterministic, i.e., that $D = \lambda L$. In this section, we'll use this relationship to compare the optimal (r, Q) parameters and their resulting expected cost to those of the EOQB model, and then to prove a bound on the worst-case error that can result from the EOQB approximation. Throughout this section, a subscript d denotes the deterministic model, i.e., the EOQB.

Since $D = \lambda L$, the inventory cost rate (5.5) simplifies to

$$g_d(y) = h(y - \lambda L)^+ + p(\lambda L - y)^+. \tag{5.38}$$

$g_d(y)$ is minimized by $S_d^* = \lambda L$ and $g_d(S_d^*) = 0$. This is not surprising: If the demand is deterministic, the inventory cost (i.e., the noncontrollable cost) equals 0 if the inventory position is kept equal to the lead-time demand. The functions $g_d(y)$ and $g(y)$, and their minimizers, are plotted in Figure 5.6.

Note that

$$g_d(y) \le g(y) \tag{5.39}$$

for all $y > 0$ (Problem 5.9). Moreover, $g(y)$ approaches $g_d(y)$ asymptotically as $y \to \pm\infty$: As $y \to +\infty$, each additional unit of inventory position (y) will almost certainly not be demanded and will therefore result in an additional unit of on-hand inventory, at a cost of h. Similarly, as $y \to -\infty$, each reduction of one unit in y will almost certainly lead to one additional stockout, at a cost of p.

Let $g_d(r, Q)$, $r_d(Q)$, $G_d(Q) = g_d(r_d(Q), Q)$, and $H_d(Q)$ be the deterministic-model versions of $g(r, Q)$, $r(Q)$, $G(Q)$, and $H(Q)$, respectively; that is, they are defined by (5.7), (5.9), (5.30), and (5.31) but with $g_d(Q)$ substituted for $g(Q)$. (See Figure 5.7.) We have

$$r_d(Q) = \lambda L - \frac{h}{h+p}Q \tag{5.40}$$

$$H_d(Q) = g_d(r_d(Q)) = \frac{hp}{h+p}Q. \tag{5.41}$$

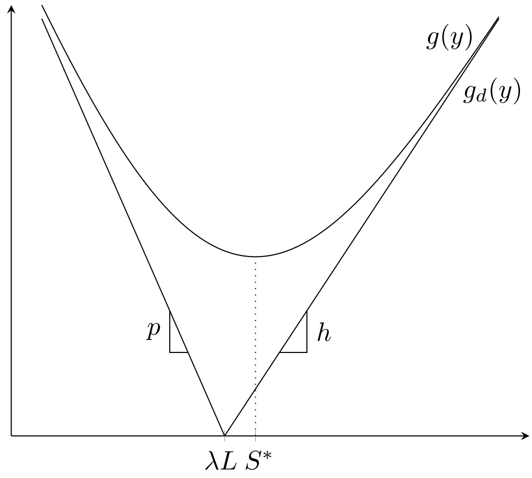


Figure 5.6 $g(Q)$ and $g_d(Q)$.

Let (r_d^*, Q_d^*) minimize $g_d(r, Q)$; from Theorem 3.5, we know that

$$r_d^* = \lambda L - \frac{h}{h+p} Q_d^* \tag{5.42}$$

$$Q_d^* = \sqrt{\frac{2K\lambda(h+p)}{hp}}. \tag{5.43}$$

In fact, one can derive (5.43) and the other two equations in Theorem 3.5 using the analysis given so far in this section, treating the EOQB explicitly as a special case of an (r, Q) policy. (See Problem 5.14.)

Theorem 5.5 $Q_d^* \le Q^* \le Q_0$*, where* Q_0 *is the* Q *that satisfies* $QH_0(Q) = 2K\lambda$*.*

Proof. Let $A_d(Q)$ be the deterministic-model version of $A(Q)$. One can show (see Problem 5.10) that $A(Q) \le A_d(Q)$ for any $Q > 0$. In particular, this holds for $Q = Q_d^*$, so

$$A(Q_d^*) \le A_d(Q_d^*) = K\lambda = A(Q^*),$$

where the two equalities follow from Theorem 5.4. Since $A(Q)$ is monotonically increasing (this can be proven rigorously but is clear from Figure 5.7), $Q_d^* \le Q^*$.

We omit the proof of the upper bound on Q^*; see Problem 5.11. ■

The fact that $Q_d^* \le Q^*$ is also evident from Figure 5.7. The upper bound of Q_0 does not provide much intuition but does provide a useful upper bound for an iterative search for Q^*, as in Algorithm 5.2.

Let $G^* = G(Q^*)$ be the optimal cost in the stochastic model, $G_0^* = G_0(Q^*)$ be the optimal controllable cost in the stochastic model, and $G_d^* = G_d(Q_d^*)$ be the optimal cost in the deterministic model. The following theorem sheds light on the relationships among these costs. The last inequality of the theorem is especially impressive, since it succinctly relates the optimal costs and solutions of the three most fundamental inventory models: the EOQ(B), the newsvendor problem, and an (r, Q) policy!

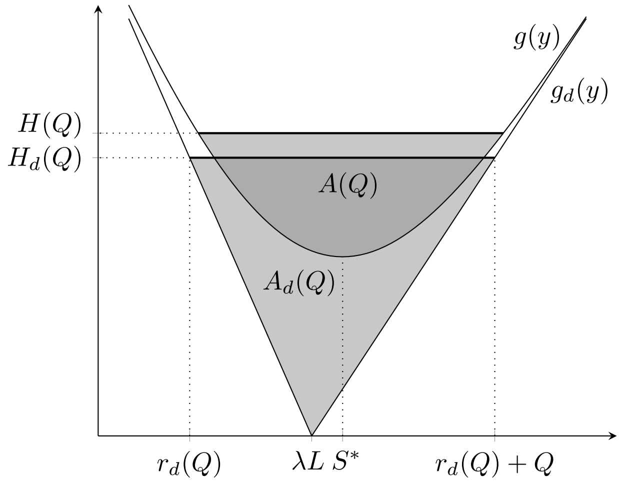


Figure 5.7 $A(Q)$ and $A_d(Q)$.

Theorem 5.6

$$\frac{Q^*}{Q_d^*}G_0^* \le G_d^* \le G^* \le g(S^*) + \frac{Q_d^*}{Q^*}G_d^*$$

Proof. First note that

$$\begin{aligned}
g(r(Q^*), Q^*) &= g(r(Q^*)) && \text{(by Theorem 5.3)}\\
\Longrightarrow\ G(Q^*) &= H(Q^*) && \text{(by definitions of } G(\cdot), H(\cdot))\\
\Longrightarrow\ G_0(Q^*) &= H_0(Q^*) && \text{(by (5.37) and (5.36))}\\
\Longrightarrow\ G_0^* &= H_0(Q^*) && \text{(by definition of } G^*) \qquad (5.44)
\end{aligned}$$

In addition, since $QH_0(Q)$ is monotonically increasing and by Theorem 5.5,

$$Q^* H_0(Q^*) \le 2K\lambda. \tag{5.45}$$

We prove the first inequality first:

$$\begin{aligned}
Q^* G_0^* &= Q^* H_0(Q^*) && \text{(by (5.44))}\\
&\le 2K\lambda && \text{(by (5.45))}\\
&= \sqrt{\frac{2K\lambda(h+p)}{hp}} \times \frac{hp}{h+p}\sqrt{\frac{2K\lambda(h+p)}{hp}}\\
&= Q_d^* H_d(Q_d^*) && \text{(by Theorem 3.5)}\\
&= Q_d^* G_d^* && \text{(by (5.44))}\\
\Longrightarrow\ \frac{Q^*}{Q_d^*}G_0^* &\le G_d^*.
\end{aligned}$$

Next, we prove the remaining two inequalities:

$$G_d^* = g_d(r_d(Q_d^*), Q_d^*) \qquad \text{(by definition of } G_d^*)$$

$$
\begin{aligned}
&\le g_d(r(Q^*), Q^*) && \text{(since } (r_d(Q_d^*), Q_d^*) \text{ are optimal for } g_d(r,Q)) \\
&= \frac{K\lambda + \int_{r(Q^*)}^{r(Q^*)+Q^*} g_d(y)dy}{Q^*} && \text{(by definition of } g_d(r,Q)) \\
&\le \frac{K\lambda + \int_{r(Q^*)}^{r(Q^*)+Q^*} g(y)dy}{Q^*} && \text{(by (5.39))} \\
&= G^* && \text{(by definition of } G^*) \\
&= g(S^*) + G_0^* && \text{(by (5.37))} \\
&\le g(S^*) + \frac{Q_d^*}{Q^*} G_d^* && \text{(by first } \le \text{ in theorem).}
\end{aligned}
$$

■

The sensitivity analysis result for the EOQ model (Theorem 3.2) also applies to the EOQB (see Problem 3.14); converted to the notation in this section, we get

$$
\frac{G_d(Q)}{G_d^*} = \frac{1}{2}\left(\frac{Q_d^*}{Q} + \frac{Q}{Q_d^*}\right).
$$

The cost function turns out to be even flatter (with respect to Q) for (r, Q) policies:

Theorem 5.7 *For any* $Q > 0$,

$$
\frac{G(Q)}{G^*} \le \frac{1}{2}\left(\frac{Q^*}{Q} + \frac{Q}{Q^*}\right).
$$

Proof. Omitted; see Zheng (1992). ■

The question now is, how accurate is the EOQB approximation? Zheng (1992) proves a fixed worst-case bound of $\frac{1}{8} = 12.5\%$ on the error that results from using the EOQB solution:

Theorem 5.8

$$
\frac{G(Q_d^*) - G^*}{G^*} \le \frac{1}{8} - \frac{1}{2}\left(\frac{1}{2} - \frac{Q_d^*}{Q^*}\right)^2 \le \frac{1}{8}
$$

Proof. Since $Q_d^* \le Q^*$ (Theorem 5.5), and $H(\cdot)$ is an increasing function,

$$
\frac{1}{Q_d^*}\int_0^{Q_d^*} H(y)dy \le \frac{1}{Q^*}\int_0^{Q^*} H(y)dy.
$$

Therefore,

$$
\begin{aligned}
G(Q_d^*) - G^* &= \frac{K\lambda + \int_0^{Q_d^*} H(y)dy}{Q_d^*} - \frac{K\lambda + \int_0^{Q^*} H(y)dy}{Q^*} \\
&\le K\lambda\left(\frac{1}{Q_d^*} - \frac{1}{Q^*}\right) \\
&= \frac{1}{2}\frac{hp}{h+p}(Q_d^*)^2\left(\frac{1}{Q_d^*} - \frac{1}{Q^*}\right).
\end{aligned} \tag{5.46}
$$

On the other hand,

$$G^* = H(Q^*) \geq H_d(Q^*) = \frac{hp}{h+p} Q^*, \tag{5.47}$$

where the first equality follows from Theorem 5.3, the inequality is proven in Problem 5.15, and the second equality follows from (5.41). Combining (5.46) and (5.47), we have

$$\begin{aligned}
\frac{G(Q_d^*) - G^*}{G^*} &\leq \frac{\frac{1}{2}\frac{hp}{h+p}(Q_d^*)^2\left(\frac{1}{Q_d^*} - \frac{1}{Q^*}\right)}{\frac{hp}{h+p}Q^*} \\
&= \frac{1}{2}\frac{(Q_d^*)^2}{Q^*}\left(\frac{1}{Q_d^*} - \frac{1}{Q^*}\right) \\
&= \frac{1}{2}\left(\frac{Q_d^*}{Q^*} - \left(\frac{Q_d^*}{Q^*}\right)^2\right) \\
&= \frac{1}{8} - \frac{1}{2}\left(\frac{1}{2} - \frac{Q_d^*}{Q^*}\right)^2 \leq \frac{1}{8}.
\end{aligned}$$

■

Like many worst-case error bounds, the bound in Theorem 5.8 overestimates the actual error bound obtained in practice. Zheng (1992) reports that, in computational results, the actual gap was less than 1% for 80.0% of the instances tested and less than 2% for 96.3%, with a maximum gap of only 2.9%. Table 5.1 reports similar results.

This raises the question of whether $\frac{1}{8}$ is the best possible bound. The answer is no: Axsäter (1996) proves that the error is no more than $(\sqrt{5} - 2)/2$, or 11.8%. This bound is tight, in the sense that there are instances whose error comes arbitrarily close to $(\sqrt{5}-2)/2$, but these instances use pathological demand distributions that do not resemble real inventory systems.

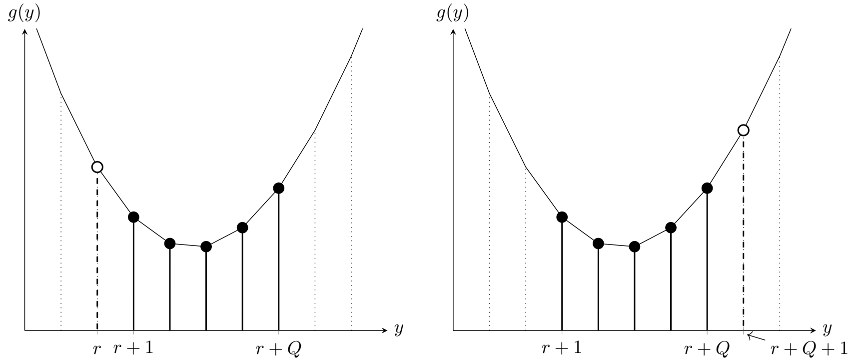
5.5 EXACT (r, Q) PROBLEM WITH DISCRETE DISTRIBUTION

Suppose now that the demand is discrete: Individual customers arrive randomly, each demanding one unit of the product. The number of demands in 1 year has a Poisson distribution with rate λ. Consequently, the lead-time demand D has a Poisson distribution with rate λL; the random variable D has pmf f and cdf F.

Since an order is placed immediately when IP reaches r, $IP \in \{r+1, r+2, \ldots, r+Q\}$ at any time. As in the model with continuous demands in Section 5.2, the inventory position spends equal time in each of these states: IP has a discrete uniform distribution on the integers $r+1, \ldots, r+Q$, so $\mathbb{P}(IP = y) = 1/Q$ for all $y = r+1, \ldots, r+Q$. (See, e.g., Zipkin (2000) for a proof.) A discrete version of the conservation-of-flow equations (4.41) and (4.43) hold, so when $IP(t) = y$, inventory (holding and stockout) costs accumulate at a rate of $g(y)$, given by (5.5) using the discrete distribution for D. Therefore, the expected total cost per year is given by

$$g(r, Q) = \frac{K\lambda + \sum_{y=r+1}^{r+Q} g(y)}{Q}, \tag{5.48}$$

which is the discrete analogue of (5.7). As before, the function $g(r, Q)$ is jointly convex in Q and r.



(a) $r(Q+1) = r(Q) - 1$. (b) $r(Q+1) = r(Q)$.

Figure 5.8 Determining which $Q+1$ y-values are optimal given $r(Q)$.

Suppose we fix Q and we want to find $r(Q)$, the best r for that Q. To do this, we need to choose r so that $g(r+1), \ldots, g(r+Q)$ are as small as possible. In other words, we want to find the Q best inventory positions $\{r+1, \ldots, r+Q\}$ to minimize the sum in (5.48). Since $g(y)$ is convex, these Q best inventory positions are nested, in the sense that, if $\{r+1, \ldots, r+Q\}$ is optimal for Q, then either $\{r, r+1, \ldots, r+Q\}$ or $\{r+1, \ldots, r+Q, r+Q+1\}$ is optimal for $Q+1$.

Figure 5.8 depicts these nested inventory positions. The solid vertical lines represent the inventory positions $r+1, \ldots, r+Q$ that are optimal for Q, while the dashed lines represent possible inventory positions to add for $Q+1$. The question is, which is the better inventory position to add, $r(Q)$ (as in Figure 5.8(a)) or $r(Q)+Q+1$ (Figure 5.8(b))? If $g(r) < g(r+Q+1)$, then we set $r(Q+1) = r(Q) - 1$; otherwise, $r(Q+1) = r(Q)$.

Note that if $Q = 1$, then (5.48) simplifies to

$$g(r, 1) = K\lambda + g(r+1). \tag{5.49}$$

The first term is a constant, so $g(r,1)$ is optimized by optimizing $g(r+1)$. From Theorem 4.3, S^*, the minimizer of $g(\cdot)$, is the smallest S such that

$$F(S) \geq \frac{p}{p+h}, \tag{5.50}$$

and the optimal r is given by

$$r = S^* - 1.$$

In other words, whenever the inventory position falls to $S^* - 1$ or smaller, we order up to S^*. This is exactly a base-stock policy under discrete demand. Thus, under discrete demand and continuous review, a base-stock policy is a special case of an (r, Q) policy.

We can find the optimal Q and r recursively, as follows. We start with $Q = 1$ and set $r(Q) = S^* - 1$, where S^* optimizes $g(S-1, 1) = K\lambda + g(S)$ from (5.49), i.e., where S^* is the smallest S satisfying (5.50). We then iterate through consecutive integer values of Q, determining $r(Q+1)$ using $r(Q)$ as described above. Since $g(r,Q)$ is convex in Q, we can stop as soon as we find that $g(r(Q+1), Q+1) > g(r(Q), Q)$. This algorithm was introduced by Federgruen and Zheng (1992). Pseudocode is given in Algorithm 5.3.

Algorithm 5.3 Exact algorithm for continuous-review (r, Q) policy with discrete demand distribution (Federgruen and Zheng 1992)

$Q \leftarrow 1$; $r(Q) \leftarrow S^* - 1$, where S^* minimizes $g(y)$ ▷ Initialization
Calculate $g(r(Q), Q)$ from (5.48)
`done` $\leftarrow$ `FALSE`
while not `done` **do** ▷ Main loop
    if $g(r(Q)) < g(r(Q) + Q + 1)$ **then** ▷ Choose $r(Q + 1)$
        $r(Q + 1) \leftarrow r(Q) - 1$
    else
        $r(Q + 1) \leftarrow r(Q)$
    end if
    Calculate $g(r(Q + 1), Q + 1)$ from (5.48)
    if $g(r(Q + 1), Q + 1) > g(r(Q), Q)$ **then** ▷ Termination check
        `done` $\leftarrow$ `TRUE`
    else
        $Q \leftarrow Q + 1$ ▷ Increment Q
    end if
end while
return $(r(Q), Q)$ ▷ Q is optimal

☐ **EXAMPLE 5.8**

Horton's Horns sells trumpets and other brass instruments. Customers arrive according to a Poisson process with a mean of 1.5 per week. Each customer demands exactly one trumpet. Horton's accountants estimate that each trumpet held in inventory costs the store \$20 per week in holding costs, and each stockout costs \$150 in penalty costs. Each order placed to the supplier incurs a fixed cost of \$100, and shipments arrive exactly 2 weeks after they are ordered. Find the optimal parameters for Horton's (r, Q) policy.

First, we have $h = 20$, $p = 150$, $K = 100$, $\lambda = 1.5$, and $L = 2$. The lead-time demand has a Poisson distribution with mean $1.5 \cdot 2 = 3$. Therefore, $g(y)$ is given by

$$g(y) = 20\bar{n}(y) + 150n(y),$$

where $n(\cdot)$ and $\bar{n}(\cdot)$ are the Poisson loss function and complementary loss function, respectively, with mean 3. (See (4.33)–(4.34).) Table 5.2 lists $F(y)$ and $g(y)$ for a range of values of y, using (C.41)–(C.42).

From (5.50), the S^* that minimizes $g(S)$ (and therefore $g(S-1, 1)$) is the smallest S such that

$$F(S) \geq \frac{150}{150 + 20} = 0.8824.$$

Since $F(4) = 0.8153$ and $F(5) = 0.9161$, $S^* = 5$. (You can also confirm that this S is optimal from Table 5.2.) Therefore, we set $r(1) = S^* - 1 = 4$. From (5.48), $g(4, 1) = 1 \cdot 1.5 + g(5) = 212.89$.

Now, $g(r(Q)) = 74.29$ and $g(r(Q) + Q + 1) = 68.62$, so we set $r(2) = r(1) = 4$. From (5.48), $g(4, 2) = (1 \cdot 1.5 + g(5) + g(6))/2 = 140.75$. The cost has gone down, so we set $Q = 2$ and continue.

Table 5.2 $F(y)$ and $g(y)$ for Poisson(3) demand with $h = 0.2$, $p = 1.5$.

y	$F(y)$	$g(y)$
0	0.0498	450.00
1	0.1991	308.46
2	0.4232	192.32
3	0.6472	114.26
4	0.8153	74.29
5	0.9161	62.89
6	0.9665	68.62
7	0.9881	82.92
8	0.9962	100.90
9	0.9989	120.25
10	0.9997	140.07

Next, $g(r(Q)) = 74.29$ and $g(r(Q)+Q+1) = 82.92$, so we set $r(3) = r(2)-1 = 3$. From (5.48), $g(3,3) = (1 \cdot 1.5 + g(4) + g(5) + g(6))/3 = 118.60$. The cost has gone down again, so we set $Q = 3$. Continuing in this manner, we find that $r(4) = r(3) = 3$ with a cost of $g(3,4) = 109.68$, $r(5) = r(4) = 3$ with a cost of $g(3,5) = 107.92$, and $r(6) = r(5) - 1 = 2$ with a cost of $g(2,6) = 108.98$. Since the cost for $Q = 6$ is greater than that for $Q = 5$, $(r, Q) = (3, 5)$ is optimal, with a cost of \$107.92. □

CASE STUDY 5.1 (r, Q) Inventory Optimization at Dell

In 2004, Dell had the largest market share of any computer-systems company and was one of the fastest growing. Dell allowed its US customers to customize their computer configurations online or over the phone; it then assembled the customized machines quickly from components at a plant in Austin, Texas, aiming to ship them to the customer within 5 days. To keep costs down, Dell held very little inventory of the components needed to assemble the finished products—typically, only a few hours' worth of inventory. However, the components were mostly manufactured in Asia, with lead times of roughly 30 days. Obviously, it is impractical to receive shipments every few hours when the shipments originate overseas. Dell's solution to this problem was to require its suppliers to hold inventory in warehouses located a few miles away from the assembly plant, which could then make deliveries to the plant several times per day. These warehouses are called *revolvers*, short for "revolving inventory."

The inventory in the revolvers was owned and managed by the suppliers, and the suppliers decided when to replenish the inventory and in what quantities. Dell was concerned that the suppliers were holding too much inventory in the revolvers, even given the frequent deliveries and high service levels required. Holding costs for component inventory was high because of the components' high value as well as its high obsolescence rate: Some computer components lose up to 2% of their value per week. The inventory in the revolvers was owned by the suppliers, so Dell did not have to bear this cost directly. But excess costs anywhere in the supply chain will eventually make their way to the consumer, which is a direct concern for Dell. Moreover, Dell's agreements with the suppliers allowed it to control the safety-stock levels at the revolvers.

Therefore, Dell partnered with the Tauber Manufacturing Institute (TMI) at the University of Michigan to study ways to reduce revolver inventory and to build a spreadsheet to perform the inventory calculations. Their project is described by Kapuscinski et al. (2004); we refer the reader to their paper for further details. Their pilot project focused on one component, given the nickname XDX.

The researchers chose to model the inventory process for XDX as a continuous-review (r, Q) policy with continuous demands. They assumed that the suppliers would continue to use the same order size Q they had been using (since Dell did not have direct control over Q) and focused on optimizing r. They assumed that the lead time (from the suppliers' manufacturing facilities in Asia to the revolvers in Texas) followed a normal distribution, as did the forecast error of the demand. (If available, the standard deviation of the forecast error is a better measure than the standard deviation of the demand when setting safety stocks; see Section 4.3.2.7.)

Since Q is known, the optimal reorder point is the r that satisfies $g(r) = g(r+Q)$, by Lemma 5.2. Of course, the model in Section 5.2 assumes deterministic lead times, but it can, in some circumstances, be applied to systems with stochastic lead times (Zipkin 1986b). However, solving $g(r) = g(r + Q)$ is not straightforward in a spreadsheet, whether the lead times are stochastic or deterministic. Therefore, the team opted instead to use a type-1 service level constraint. From (5.21) and (5.25), the optimal reorder point is given by

$$r = \mu_L \lambda + z_\alpha \sqrt{\lambda^2 \sigma_L^2 + \mu_L \sigma_e^2}, \tag{5.51}$$

where μ_L and σ_L are the mean and standard deviation of the lead time, λ is the mean forecasted demand per day during the upcoming lead time, σ_e is the standard deviation of the daily forecast error during the upcoming lead time, and α is the desired type-1 service level. The first term of (5.51) is the cycle stock (most of which represents in-transit inventory from the supplier overseas) and the second is the safety stock. The parameters μ_L, σ_L, λ, and σ_e can be updated daily based on new observed data and forecasts of the near future, resulting in new calculations of r.

The key remaining question is how to determine α. The researchers chose to set it equal to the critical ratio: $\alpha = p/(p + h)$, where p and h are the the stockout and holding costs per unit per day. These parameters, in turn, were estimated using a combination of historical data and subjective opinions. The stockout cost p included estimates of lost profit from a canceled order and expedited shipping costs, as well as the probability that each unmet demand would result in either cancellation or expediting. The holding cost h included estimates of the supplier's cost of capital, price erosion due to obsolescence, and physical storage costs at the revolver.

The team built a user-friendly spreadsheet to manage and process the data and to perform the inventory calculations. The spreadsheet also provided charts showing the inventory levels, service levels, and so on, for the historical data for both the current and recommended inventory policies. The optimal inventory levels turned out to be fairly close to the current levels *on average*, but the team found that the current policies generated widely fluctuating inventory levels. For example, during one period, the system had nearly twice as much inventory as was required (resulting in excess costs), while during another period, the system had only about two-thirds of the required amount (resulting in excess stockout risk). Overall, the team estimated that Dell could reduce its inventory of XDX by roughly 38%, which would result in a savings of over $40 million over the life of the component.

PROBLEMS

5.1 (Exact and Approximate r and Q: Continuous Demand) Consider an (r, Q) policy for continuous demands. Suppose the annual demand is distributed $N(800, 40^2)$, the fixed cost is $K = 50$, and the holding and stockout costs are $h = 3.1$ and $p = 45$, respectively, per item per year. The lead time is 4 days. Find r and Q using each of the methods below.

- **a)** The EIL approximation.
- **b)** The EOQB approximation.
- **c)** The EOQ+SS approximation.
- **d)** The loss-function approximation.
- **e)** Algorithm 5.2 for exact optimal values of r and Q.

For each method, report the values of r and Q you found, as well as the corresponding expected annual cost from (5.7).

5.2 (Exact and Approximate r and Q: Discrete Demand) Consider an (r, Q) policy for discrete demands. Suppose the demand has a Poisson distribution with a mean of $\lambda = 12$ units/month, the fixed cost is $K = 4$, and the holding and stockout costs are $h = 4$ and $p = 28$, respectively, per item per month. The lead-time is 0.5 months.

- **a)** Find approximate values for r and Q by using the EOQB approximation described in Section 5.3.2, replacing $g(y)$ with (4.32) when solving (5.9).
- **b)** Find exact optimal values for r and Q using Algorithm 5.3.

For each method, report the values of r and Q you found, as well as the corresponding expected cost per week from (5.48).

5.3 ((r, Q) for Automobile Components) Return to the automobile manufacturing plant from Problem 3.5. Suppose now that the rate at which the plant uses power-lock mechanisms is stochastic and normally distributed, with a mean of 192 per day (8 per hour) and a standard deviation of 17.4 per day. Replenishment orders for power-lock mechanisms incur a lead time of 3 days. If the plant runs out of power locks, it must expedite them from the supplier at a cost of $40 each. Using the EIL approximation for (r, Q) policies in Section 5.3.1, find approximate values for r and Q. Also report the expected total cost per week, using equation (5.7).

- **a)** The EIL approximation.
- **b)** The EOQB approximation.
- **c)** The EOQ+SS approximation.
- **d)** The loss-function approximation.
- **e)** Algorithm 5.2 for exact optimal values of r and Q.

5.4 (Lackluster Video) Lackluster Video needs to decide how may DVD copies of the new hit movie *The Supply Chain's Weakest Link* to stock in its stores. The company expects demand for DVD rentals for the movie over the next 90 days to be Poisson with a mean of λ per day. The length of time each renter keeps a DVD before returning it is exponential with a mean of $1/\mu$ days (i.e., exponential with a rate of μ).

Each copy purchased by the store costs c. Demands are backordered, in the sense that a customer wanting to rent the movie but finding that it is out of stock will return on another day to try again. Since this movie has been designated as a "guaranteed in stock" title, each backordered demand incurs a stockout cost of g, the cost of providing a free rental to the customer.

Assuming that backordered customers check back frequently to see whether the movie is in stock and rent it quickly when it is available, this system can be modeled as an $M/M/S$ queue, where S is the number of copies of the DVD owned by the store. It can be shown that the probability of a stockout in an $M/M/S$ queue is approximately

$$P[\text{stockout}] \approx 1 - \Phi\left(\frac{S - \rho - \frac{1}{2}}{\sqrt{\rho}}\right),$$

where Φ is the standard normal cdf and $\rho = \lambda/\mu$ (in queuing terminology, the "offered load").

a) Determine the optimal number of copies to purchase (S) to minimize the purchase cost and the expected stockout cost over the next 90 days using the approximation given above. (Assume that the demand after 90 days will be negligible.) Your answer should be in closed form; that is, $S =$ [some expression].

b) Compute the optimal S assuming that $\lambda = 22$, $\mu = \frac{1}{4}$, $c = 9$, and $g = 4.5$.

c) Suppose the video store is worried about loss-of-goodwill costs as well as free rental costs when a demand is backordered, but it is uncomfortable estimating these costs. Instead, it would prefer to choose S so that demands are met with probability α. Prove that the smallest such S is given by

$$S \approx \rho + z_\alpha \sqrt{\rho}.$$

d) In two or three sentences, interpret the result from part (c) in terms of cycle and safety stock.

5.5 (Heating Oil Replenishments) Henry's Heating Oil company delivers oil to its customers' homes. If a customer signs up for Henry's "auto-fill" plan, the company delivers oil to the customer's home on a regular schedule based on historical oil-usage data for that customer. Suppose a given customer has an oil tank that holds C liters of oil. For each delivery to this customer, Henry's incurs a fixed cost of K, representing the cost of the truck, driver, and fuel required to make the delivery. Henry's will make a delivery to this customer every T days, where T is a decision variable, and at each delivery, it will deliver enough oil to fill the tank. The number of days required for the customer to use C liters of oil is a random variable, denoted X, whose pdf and cdf are f and F, respectively. If the customer uses all C liters of oil before the next delivery, Henry's must make an emergency delivery to refill the tank. For these emergency deliveries, the regular fixed cost of K does not apply, but instead Henry's incurs a penalty cost of pT. (The penalty cost is proportional to T because the more infrequent the deliveries, the more disruptive it is to

Henry's delivery schedule to add an emergency delivery.) After the emergency delivery, the regular schedule resumes; that is, the next delivery will be T days after the last *regular* delivery. Assume the customer never needs more than one emergency shipment between two regular shipments.

a) Write the expected cost per day as a function of T.

b) Find an optimality condition for the delivery interval, T. You may assume that X is normally distributed and that $T < \mathbb{E}[X]$.

c) Suppose $C = 500$, $K =$\$175, $p =$\$25, and $X \sim N(22, 8^2)$. What is T^*, and what is the corresponding expected cost per day?

5.6 **(Stockout-Constrained Service Level)** Consider the EIL approximation in Section 5.3.1. Define a new type of service level as follows: $SL(a)$ is the percentage of order cycles during which there are at most a stockouts, for constant $a \geq 0$. Suppose that we wish to enforce a service level constraint that says $SL(a) \geq \gamma$, for fixed $0 \leq \gamma < 1$. What are the optimal values of r and Q for the problem with this service level constraint?

5.7 **(Properties of $r(Q)$)** For the exact continuous (r, Q) model in Section 5.2, prove that, for any $Q > 0$:

a) $r(Q) < S^* < r(Q) + Q$

b) $-1 < r'(Q) < 0$; $r(Q)$ is decreasing; and $r(Q) + Q$ is increasing

c) $\lim_{Q\to\infty} r(Q) = -\infty$ and $\lim_{Q\to\infty} r(Q) + Q = \infty$

5.8 **(Proof of (5.32))** Prove equation (5.32).

5.9 **(Deterministic vs. Stochastic Inventory Cost Rate)** Prove that $g_d(y) \leq g(y)$ for all $y > 0$, where $g_d(y)$ is defined in (5.38) and $g(y)$ is defined in (5.5).

5.10 **(Deterministic vs. Stochastic $A(Q)$)** Prove that, for any $Q > 0$, $A(Q) \leq A_d(Q)$, where $A(Q)$ is defined in (5.34) and $A_d(Q)$ is its deterministic-model analogue.

5.11 **(Proof of Upper Bound on Q^*)** Complete the proof of Theorem 5.5 by proving that $Q^* \leq Q_0$.

5.12 **(Range of Q^* Bounds as K Changes)** By Theorem 5.5, Q^* is contained in the interval $[Q_d^*, Q_0]$, where Q_0 satisfies $QH_0(Q) = 2K\lambda$. In this problem, you will prove that the width of this interval is bounded by a constant for all $K > 0$. (On the other hand, the constant will change as the other cost parameters change.)

a) Let Q_1 be the Q that satisfies $H_0(Q) = H_d(Q_d^*)$. Prove that $Q_0 \leq Q_1$.

b) Prove that $H'(Q) > 0$ for all $Q > 0$ and that $\lim_{Q\to\infty} H'(Q) = hp/(h + p)$.

c) Prove that $Q_1 - Q_d^*$ is an increasing function of K and converges to a constant as $K \to \infty$.

Hint: Argue that it is sufficient to prove the result with respect to increases in Q_d^* rather than K.

d) Prove that $Q_0 - Q_d^*$ is bounded by a constant for all $K > 0$.

You may use the properties in Problem 5.7 without proof.

5.13 **(EOQB Error Vanishes as $K \to \infty$)** Using the analysis in Section 5.4.3, prove that $(G(Q_d^*) - G^*)/G^* \to 0$ as $K \to \infty$.

5.14 (EOQB as Special Case of (r, Q)) Prove Theorem 3.5 by treating the EOQB as a special case of an (r, Q) policy, using the analysis in Section 5.4.3.

5.15 ($H(Q)$ vs. $H_d(Q)$) Using the analysis in Section 5.4.3, prove that $H_d(Q) \leq H(Q)$ for all $Q > 0$.

5.16 (Lead-Time Demand under Stochastic Lead Times) Prove equations (5.24) and (5.25).

5.17 (No Fixed Bound for r_d^*) In the exact (r, Q) model, suppose we set $Q = Q_d^*$ as in Section 5.2, but we set $r = r_d^*$ instead of $r = r(Q_d^*)$. Prove that there is no fixed worst-case bound for this approach.

5.18 (No Fixed Bound for EOQ+SS Approximation) Prove that there is no fixed worst-case error bound for the EOQ+SS approximation for the optimal (r, Q) policy.

5.19 (Joe's Corner Store with Poisson Demand) Suppose that Joe's Corner Store from Example 5.2 faces Poisson annual demand with a mean of 1300. Using Algorithm 5.3, find r^*, Q^*, and $g(r^*, Q^*)$.

5.20 ((r, Q) with Minimum Order Quantity) Suppose that $K = 0$ but there is a minimum order quantity constraint that requires that $Q \geq Q_{\min}$ for some constant $Q_{\min}$. Assume the demand has a discrete distribution. Explain how to modify Algorithm 5.3 to handle this case.

5.21 (Solution in Terms of Standard Normal) In this problem, you will investigate what happens to (r^*, Q^*) and $g(r^*, Q^*)$ in the exact model (Section 5.2) as the lead-time demand parameters μ and σ change. In particular, you will investigate the relationship between the solution under $N(\mu, \sigma^2)$ demand and that under $N(0, 1)$ demand.

Assume that $\sigma^2 = \lambda/\lambda_0$ for some constant λ_0 but that μ can vary independently of σ^2 and λ.

Let $g_0(r, Q)$ be the expected cost function of the exact model under $N(0, 1)$ lead-time demand. Let (r_0, Q_0) be the optimal parameters for this system and g_0^* be the optimal cost; that is,

$$g_0^* = g_0(r_0, Q_0) = \min_{r,Q} g_0(r, Q).$$

Similarly, let (r^*, Q^*) be the optimal parameters for the system with $N(\mu, \sigma^2)$ lead-time demand, and let $g^* = g(r^*, Q^*)$.

Prove that

$$r^* = \mu + r_0\sigma \tag{5.52}$$

$$Q^* = Q_0\sigma \tag{5.53}$$

$$g^* = g_0^*\sigma. \tag{5.54}$$

5.22 (Bound on Q^*) Let Q^* be the optimal order quantity for the exact model with continuous demands in Sections 5.2 and 5.4, and let Q_d^* be the optimal order quantity for the EOQB. Let

$$Q_\sigma = g(S^*)\frac{h + p}{hp}.$$

(Q_σ does not have a precise interpretation. But it is, in a sense, a quantity for the newsvendor model that is analogous to Q_k for the EOQB, since in the EOQB, the optimal order quantity equals the optimal cost times $(h+p)/hp$.)

Prove that

$$Q^* \le Q_\sigma + Q_d^*.$$

Hint: First prove that

$$H(Q) \ge \frac{hp}{h+p} Q$$

for all $Q > 0$. (You may use the result of Problem 5.15 without proof.) Then use this to prove the result.

5.23 (Stockout Cost without SA2) Suppose we do not assume SA2. Show that the expected stockout cost per year under the EIL approximation has $(d-r)^2$ in the integrand instead of $(d-r)$.

5.24 (EIL Approximation with One-Time Stockout Cost) Consider an inventory system that functions almost exactly like the system described in Section 5.3.1 on the EIL approximation for the (r, Q) problem. The only difference is that, when we run out of inventory, the stockout cost p is incurred immediately, and only once, regardless of how many demands occur before the replenishment order arrives from the supplier.

a) Formulate the objective function $g(r, Q)$, analogous to (5.16).

b) Identify optimality conditions for Q and r, similar to equations (5.17) and (5.18). Your optimality conditions do *not* need to be in closed form, i.e., they do not need to look like $Q = \cdots$ or $r = \cdots$.

CHAPTER 6

MULTIECHELON INVENTORY MODELS

6.1 INTRODUCTION

In this chapter, we study inventory optimization models for multiechelon (or multistage) systems with shipments made among the stages. There are two common ways to interpret the stages or nodes in a multiechelon system:

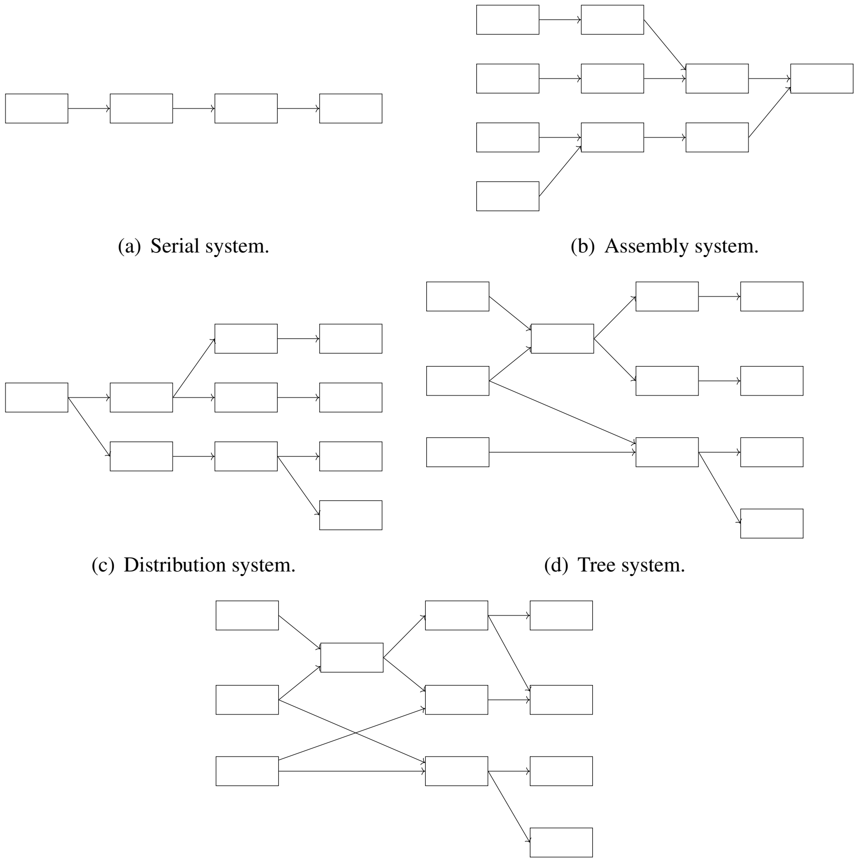
1. Stages represent locations in a supply chain network, and links among the stages represent physical shipments of goods. For example, the stages in Figure 6.1(a) may represent the following physical locations: a supplier in China, a factory in California, a warehouse in Chicago, and a retailer in Detroit (respectively).

2. Stages represent processes that the product must undergo during manufacturing, assembly, and/or distribution. Links among the stages represent transitions between steps in the process. For example, the stages in Figure 6.1(a) may represent the following processes: manufacturing, assembly, testing, and packaging. These four functions may take place in four different locations or all within the same building—it is largely irrelevant from the perspective of the model. We sometimes refer to the stages as different "products," even if they really represent different phases of producing a single product.

Either interpretation is acceptable for the models that we discuss, although some models are more naturally interpreted in one way than the other. In the discussion that follows,

Fundamentals of Supply Chain Theory, Second Edition. Lawrence V. Snyder and Zuo-Jun Max Shen.

Companion website: www.wiley.com/go/Snyder/SupplyChainTheory



(a) Serial system. (b) Assembly system. (c) Distribution system. (d) Tree system. (e) General system.

Figure 6.1 Multiechelon network topologies.

we will use terms such as “shipped” or “transferred” under either interpretation to mean “moved from one stage to the next.”

6.1.1 Multiechelon Network Topologies

Multiechelon networks can be structured in a number of ways, and the network’s topology plays a large role in determining how the system is analyzed and optimized. The simplest multiechelon topology is a *serial system* (or *series system*), in which each echelon contains exactly one stage. Put another way, every stage has exactly one predecessor and exactly one successor, except for two stages, one of which has exactly one successor and no predecessors, and the other of which has exactly one predecessor and no successors. (A *predecessor* of stage j is another stage that ships product to j, and a *successor* of j is another stage that j ships to.) See Figure 6.1(a) for an example of a serial system.

In an *assembly system*, each stage has at most one successor; see Figure 6.1(b). Interpretation (2) is most common for assembly systems: The network represents a *bill-of-materials* structure that describes how a final product is assembled from raw materials and intermediate products. In this case, the links in the network indicate "and" relationships: To make one unit of the product at stage j, we need one (or more) unit of each of j's predecessors. Assembly systems can also be viewed under interpretation (1), with links denoting the geographic flow of materials. If stage j has three predecessors, then there are three stages that make the product and ship it to stage j. Here, too, links represent "and" relationships since all three upstream stages ship product to stage j. An alternate, but less common, way to use interpretation (1) is that the links represent "or" relationships, and stage j's predecessors are multiple alternate suppliers from which stage j can order. In a given order cycle, it may order from one, more than one, or all of its predecessors, depending on their capacities, the observed demands, and so on. Under any of these interpretations, assembly systems are commonly used to model upstream portions of supply chains whose purpose is to consolidate products or locations into a few stages.

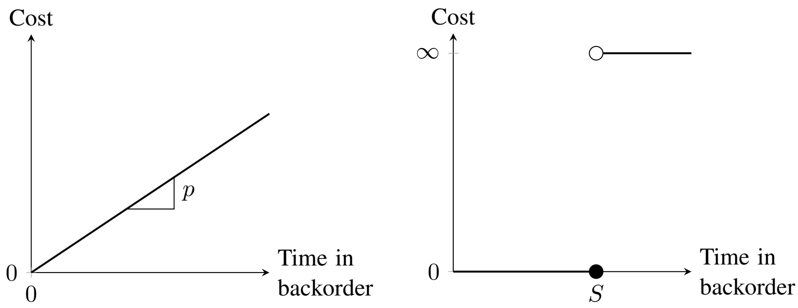
A *distribution system* (Figure 6.1(c)) is the opposite of an assembly system: Each stage has at most one predecessor. Interpretation (1) is most common for distribution systems, which are often used to model downstream portions of supply chains—the portion that moves material from a few centralized locations to a set of retailers or customers distributed throughout a large geographical region.

Tree systems (Figure 6.1(d)) are hybrids of assembly and distribution systems—each stage may have multiple predecessors and successors—but tree systems may contain no undirected cycles. (A *cycle*, in graph theory, is a portion of the graph whose links allow one to move from a starting node, through a sequence of other nodes, and back to the starting node, without repeating any other nodes links. An *undirected cycle* is a cycle in the graph that results from removing all of the arrows from the links so that movement can go in either direction.) Finally, *general systems* allow any number of successors and predecessors and have no restrictions on undirected cycles. Figure 6.1(e) shows an example. General systems are the most flexible topology but are also the most difficult to analyze and optimize.

6.1.2 Stochastic vs. Guaranteed Service

The most challenging aspect of multiechelon inventory models is that a given stage j provides stochastic lead times to its successors, even if the transportation lead time is deterministic, due to occasional stockouts at stage j, and the optimal inventory parameters at stage j's successors depend on the probability distributions of these stochastic lead times. We have discussed some results for optimizing single-stage systems with stochastic lead times (see, e.g., Section 5.3.3), but in those models, we assume the lead-time distributions are known and that the lead times are iid. In contrast, the probability distributions of the lead times in multiechelon systems are very complex and difficult to characterize, the lead times are not iid, and moreover, the distributions depend on the upstream inventory parameters. Even for single-stage systems, the distributions of the lead times generated by the stage are quite complex (Higa et al. 1975, Sherbrooke 1975).

Two primary types of models have been developed to handle these complexities in multiechelon base-stock systems: stochastic-service models and guaranteed-service models (Graves and Willems 2003a). In *stochastic-service models*, each stage i sets a base-stock level S_i and meets demand from stock whenever possible using this base-stock level. The actual lead time seen by downstream stages is stochastic since some demands will be backordered. This is the approach taken in the seminal model of Clark and Scarf (1960) and related works, discussed in Section 6.2.



(a) Stochastic-service models. (b) Guaranteed-service models.

Figure 6.2 Interpretation of stockout penalties.

In *guaranteed-service models*, stage i sets a "committed service time" (CST), denoted S_i, within which it is required to satisfy *every* demand.[1] For example, if S_i =5 periods, then every demand must be satisfied in no more than 5 periods. To make this guarantee, guaranteed-service models require the demand to be bounded above. The guaranteed-service assumption provides the strategic safety stock placement problem (SSSPP), described in Section 6.3, its tractability. There is a close relationship between the CST and the base-stock level, since a larger base-stock level allows the stage to quote a shorter CST. In fact, any given set of CSTs implies a certain set of base-stock levels, and the base-stock levels, not the CSTs, are usually the main quantities of interest.

One way to view the difference between these two approaches is that guaranteed-service models allow a CST of $S > 0$ but require a service level of $\alpha = 1$ while stochastic-service models assume a service time of $S = 0$ but allow a less restrictive service level of $\alpha < 1$. Another interpretation is that stockouts in stochastic-service models incur a penalty that is proportional to the time the unit is backordered, whereas in guaranteed-service models, no penalty is incurred until the backorder has lasted S periods, and after that the penalty is ∞. (See Figure 6.2.) In fact, in guaranteed-service models, a backorder isn't really even considered a backorder until it has lasted S periods.

It is important to remember that these are both merely mathematical models, two different ways to describe the mechanics and the optimization problem underlying a multiechelon inventory system. The end result of either approach is a set of base-stock levels, even though the decision variables in the guaranteed-service model are the CSTs rather than base-stock levels. Thus, the guaranteed-service model can be used even when stages do not actually quote CSTs to one another. Either modeling approach can be used to model a given system, and the choice of approach is a modeling decision with pros and cons just like any other.

In Section 6.2, we first discuss stochastic-service models, describing an optimal and a heuristic approach for optimizing base-stock levels in serial systems and then briefly discussing the extent to which these methods can be extended to solve assembly and dis-

[1]Unfortunately, the literature on stochastic-service models and that on guaranteed-service models have both laid claim to the notation S, but they use it to mean very different things. In stochastic-service models, S denotes the base-stock level, whereas in guaranteed-service models, S denotes the CST. We have opted to use S for both purposes to remain consistent with these two bodies of literature, at the risk of confusing the reader. It is safe to assume that S denotes a base-stock level in Section 6.2 and a CST in Section 6.3.

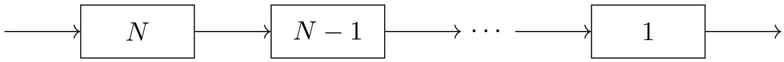


Figure 6.3 N-stage serial system in stochastic-service model.

tribution systems. Then, in Section 6.3, we discuss guaranteed-service models, beginning with an analysis of single-stage systems and working our way up to tree systems.

See van Houtum et al. (1996) and Graves and Willems (2003a) (among others) for further reviews of the literature on stochastic- and guaranteed-service models, respectively.

6.2 STOCHASTIC-SERVICE MODELS

6.2.1 Serial Systems

Consider an N-stage serial system, with the stages labeled as in Figure 6.3. Stage 1 is farthest downstream. It faces stochastic external customer demand and places replenishment orders to stage 2, which places replenishment orders to stage 3, and so on up the line to stage N. Stage N, in turn, places replenishment orders to an external supplier that is assumed to have infinite supply.

We consider a continuous-review, infinite-horizon system, though nearly all of the results described below hold (with slight modifications) for periodic-review systems, as well. Orders placed by stage j incur a transportation lead time of L_j; that is, the order is received L_j time units later if stage $j+1$ had sufficient stock to ship the order immediately, and more than L_j time units later otherwise. Stage j incurs a holding cost of h'_j per item per time unit, which is charged on the on-hand inventory at stage j as well as on the inventory in transit to stage $j-1$. (One can show that the expected number of units in transit is a constant, and therefore the in-transit holding cost does not affect the optimization.) Unmet demands are backordered at all stages, but only stage 1 incurs a stockout cost, given by p per item per time unit. There are no fixed costs, and we will ignore any per-unit ordering costs.

In multiechelon inventory theory, the *echelon* of stage j (or just "echelon j") is defined as the set of stages $\{j, j-1, \ldots, 1\}$; that is, the set that includes j and all downstream stages. Note that this is a particular inventory-theoretic use of the term "echelon" and is different from the way we defined it in Chapter 1. Stage j's *echelon inventory* is the total inventory in echelon j, and its *local inventory* is the inventory at stage j only. It turns out to be more convenient to optimize stage j's echelon inventory rather than its local inventory.

Stage j's *local on-hand inventory*, denoted I'_j, includes items on hand at stage j only, whereas stage j's *echelon on-hand inventory*, denoted I_j, includes all of the on-hand inventory in echelon j, plus all of the in-transit inventory among these stages:

$$I_j = \sum_{i=1}^{j} (I'_i + IT_{i-1}), \tag{6.1}$$

where IT_{i-1} is the inventory in transit from i to $i-1$, and $IT_0 \equiv 0$. Stage j's *local* and *echelon inventory levels*, denoted IL'_j and IL_j, respectively, are given by

$$IL'_j = I'_j - B'_j \tag{6.2}$$

$$IL_j = I_j - B'_1, \tag{6.3}$$

where B'_1 is the (local) backorders at stage 1. Note that the local inventory level at stage j subtracts the backorders at stage j while the echelon inventory level subtracts those at stage 1; upstream backorders are not counted in IL_j, and therefore the echelon inventory level does not equal the sum of the local quantities.

The holding cost h'_j is called a *local holding cost*, and it is charged based on the number of items in stage j's local inventory plus the number of items in transit from stage j to $j-1$, IT_{j-1}. We will mostly work with stage j's *echelon holding cost*, denoted h_j and defined as

$$h_j = h'_j - h'_{j+1} \tag{6.4}$$

(with $h'_{N+1} \equiv 0$). Typically, local holding costs increase as we move downstream in the supply chain since value is added to the product at each stage. Therefore, h_j represents the holding cost corresponding to the value added at stage j. It turns out that we can calculate total holding costs using either echelon or local quantities:

Proposition 6.1

$$\sum_{j=1}^{N} h_j I_j = \sum_{j=1}^{N} h'_j \left(I'_j + IT_{j-1} \right), \tag{6.5}$$

where $IT_0 \equiv 0$.

Proof. Omitted; see Problem 6.4. ■

The following theorem establishes the form of the optimal inventory policy for serial systems. It was proved for finite-horizon problems in the seminal paper of Clark and Scarf (1960) and for infinite-horizon problems by Federgruen and Zipkin (1984).

Theorem 6.2 *An echelon base-stock policy is optimal at each stage of a serial system with no fixed costs.*

In an *echelon base-stock policy*, each stage j has a fixed level S_j, called the *echelon base-stock level*, and it places an order as needed to bring its *echelon inventory position* (defined as stage j's echelon inventory level, IL_j, plus any items on-order from stage $j+1$) equal to S_j. An echelon base-stock policy is essentially the same as the base-stock policies we are already familiar with except that it is the echelon inventory, rather than the local inventory, that we compare to the base-stock level when making ordering decisions. We use $\mathbf{S} = (S_j)_{j=1}^{N}$ to denote the vector of echelon base-stock levels, one for each stage.

We will discuss approaches for finding optimal or near-optimal echelon base-stock levels. Local base-stock levels (denoted S'_j) can be obtained from the echelon base-stock levels by setting

$$S'_j = S_j - S_{j-1}, \tag{6.6}$$

defining $S_0 \equiv 0$. (This assumes that $S_j \geq S_{j-1}$. If not, we let $S^-_j = \min_{i \geq j}\{S_i\}$ and set $S'_j = S^-_j - S^-_{j-1}$, again setting $S^-_0 \equiv 0$.) And echelon base-stock levels can be obtained from local ones as follows:

$$S_j = \sum_{i=1}^{j} S'_i. \tag{6.7}$$

Let D_j be a random variable representing the lead-time demand at stage j. Since stage j's demands are ultimately generated by the external customer (via orders placed to stage 1, then to stage 2, and so on), stage j's demand per time unit has the same distribution as the customer's demand, but the distribution of stage j's lead-time demand D_j depends on L_j. Let $F_j(\cdot)$ be the cdf of D_j.

Table 6.1 summarizes the notation for the stochastic-service model.

Table 6.1 Stochastic-service model notation summary.

Quantity	Echelon	Local
Holding cost	$h_j = h'_j - h'_{j+1}$	$h'_j = \sum_{i=j}^N h_i$
Stockout cost	p	p
Inbound lead time	L_j	L_j
On-hand inventory	I_j	I'_j
Backorders	—	B'_j
Inventory level	$IL_j = I_j - B'_1$	$IL'_j = I'_j - B'_j$
On-order items	OO_j	OO_j
Inventory position	$IP_j = IL_j + OO_j$	$IP'_j = IL'_j + OO_j$
Inbound in-transit inventory	IT_j	IT_j
Inventory–transit position	$ITP_j = IL_j + IT_j$	—
Base-stock level†	$S_j = \sum_{i=1}^j S'_i$	$S'_j = S_j - S_{j-1}$
Vector of base-stock levels	$\mathbf{S} = (S_j)_{j=1}^N$	$\mathbf{S}' = (S'_j)_{j=1}^N$
Lead-time demand	D_j	D_j
cdf of lead-time demand	$F_j(\cdot)$	$F_j(\cdot)$

†Formula for S'_j assumes $S_j \geq S_{j-1}$; see page 192.

6.2.2 Exact Approach for Serial Systems

For a given set of base-stock levels, the expected cost of the system can be expressed using either local or echelon quantities:

$$g'(\mathbf{S}') = \mathbb{E}\left[\sum_{j=1}^N h'_j \left(I'_j + IT_{j-1}\right) + pB'_1\right] \tag{6.8}$$

$$g(\mathbf{S}) = \mathbb{E}\left[\sum_{j=1}^N h_j IL_j + (p + h'_1) IL_1^-\right]. \tag{6.9}$$

(Note that the prime in $g'(\cdot)$ indicates local quantities, not a derivative.) These two expressions are equivalent (see Problem 6.5), but (6.9) will be more convenient for us to work with.

We wish to choose $\mathbf{S}$ to minimize $g(\mathbf{S})$. $g(\mathbf{S})$ is a messy function of $\mathbf{S}$ because the inventory levels on the right-hand side depend on $\mathbf{S}$ in messy ways. In fact, since S_j affects the inventory levels at all stages downstream from j, it would seem that we need to jointly optimize all of the S_j simultaneously. Fortunately, a much simpler and more elegant procedure suffices.

Let ITP_j be the *echelon inventory–transit position* at stage j, which equals the echelon inventory level at j plus all items in transit from stage $j+1$:

$$ITP_j = IL_j + IT_j \tag{6.10}$$
$$= IP_j - (IL'_{j+1})^-. \tag{6.11}$$

That is, IP_j includes *all* items that have been ordered from $j+1$ but not yet received, whereas ITP_j only includes items that have been shipped. The difference between the two equals the number of backorders at $j+1$ (i.e., $(IL'_{j+1})^-$), and they are equal if there are no backorders at $j+1$.

The conservation-of-flow argument from Section 4.3.4.1 can be applied here to show that

$$IL_j(t + L_j) = ITP_j(t) - D_j, \tag{6.12}$$

since all items that were shipped from $j+1$ at or before period t have arrived by period $t + L_j$, no items that were shipped after t have arrived, and the intervening demand is D_j. This equation is similar to (4.41), except that the inventory position is replaced by the inventory–transit position.[2] In the single-stage models in Chapters 4 and 5, the supplier never has stockouts, so ITP and IP are equal.

One can show that

$$ITP_j(t) = \min\{S_j, IL_{j+1}(t)\}. \tag{6.13}$$

Intuitively, (6.13) says that the inventory at or en route to echelon j equals the echelon base-stock level at j, unless the upstream inventory is insufficient to attain the base-stock level, in which case it equals the upstream inventory level. For a more rigorous proof, see Problem 6.14.

In addition, note that at stage N,

$$IP_N(t) = ITP_N(t) = S_N \tag{6.14}$$

for all t, since the upstream supplier to stage N never has stockouts.

In steady state, we can rewrite (6.12), (6.13), and (6.14) as

$$ITP_N = S_N \tag{6.15}$$
$$IL_j = ITP_j - D_j \tag{6.16}$$
$$ITP_{j-1} = \min\{S_{j-1}, IL_j\}. \tag{6.17}$$

Equations (6.15)–(6.17) provide a recursion that expresses ITP_{j-1} in terms of IL_j, IL_j in terms of ITP_j, and so on, until we reach ITP_N, which equals a constant.

We next introduce three auxiliary functions that condition the expected cost of the system on the state variables in the recursion. These functions will allow us to develop a recursion for the (unconditional) expected cost for a given vector $\mathbf{S}$ of base-stock levels, and then to find the optimal base-stock vector.

$$\hat{g}_j(x|\mathbf{S}) = \mathbb{E}\left[\sum_{i=1}^{j} h_i IL_i + (p + h'_1)IL_1^- \middle| IL_j = x\right] \tag{6.18}$$

[2]The notation is slightly different. Here, we indicate the time index in parentheses rather than as subscripts, as is common for continuous-review systems.

$$g_j(y|\mathbf{S}) = \mathbb{E}\left[\sum_{i=1}^{j} h_i IL_i + (p+h'_1)IL_1^- \middle| ITP_j = y\right] \tag{6.19}$$

$$\underline{g}_j(x|\mathbf{S}) = \mathbb{E}\left[\sum_{i=1}^{j} h_i IL_i + (p+h'_1)IL_1^- \middle| IL_{j+1} = x\right] \tag{6.20}$$

Each auxiliary function fixes one of the recursion variables—IL_j, ITP_j, or IL_{j+1}—and then calculates the expected cost in stages $1, \ldots, j$ using that value as the starting point. For example, suppose we have a 4-stage system with base-stock vector $\mathbf{S}$ and we know that $IL_3 = x$. Then the expected cost for stages 1 and 2 is given by $\underline{g}_2(x|\mathbf{S})$. Similarly, the expected cost in stages 1 and 2 is $g_2(y|\mathbf{S})$ if we know that $ITP_2 = y$ and is $\hat{g}_2(x|\mathbf{S})$ if we know that $IL_2 = x$.

We can write (6.18)–(6.20) recursively. First let

$$\underline{g}_0(x|\mathbf{S}) = (p+h'_1)x^-.$$

Then

$$\begin{aligned}\hat{g}_1(x|\mathbf{S}) &= \mathbb{E}\left[h_1 IL_1 + (p+h'_1)IL_1^- | IL_1 = x\right] \\ &= h_1 x + \underline{g}_0(x|\mathbf{S}).\end{aligned}$$

Similarly,

$$\begin{aligned}g_1(y|\mathbf{S}) &= \mathbb{E}\left[h_1 IL_1 + (p+h'_1)IL_1^- | ITP_1 = y\right] \\ &= \mathbb{E}_{D_1}\left[\mathbb{E}\left[h_1 IL_1 + (p+h'_1)IL_1^- | IL_1 = y - D_1\right]\right] \\ &= \mathbb{E}\left[\hat{g}_1(y - D_1|\mathbf{S})\right],\end{aligned}$$

where the expectation is over D_1. (The second equality follows from (6.16).) And,

$$\begin{aligned}\underline{g}_1(x|\mathbf{S}) &= \mathbb{E}\left[h_1 IL_1 + (p+h'_1)IL_1^- | IL_2 = x\right] \\ &= \mathbb{E}\left[h_1 IL_1 + (p+h'_1)IL_1^- | ITP_1 = \min\{S_1, x\}\right] \\ &= g_1(\min\{S_1, x\}|\mathbf{S}),\end{aligned}$$

where the second equality follows from (6.17). Continuing this process, we get

$$\begin{aligned}\hat{g}_2(x|\mathbf{S}) &= \mathbb{E}\left[h_1 IL_1 + h_2 IL_2 + (p+h'_1)IL_1^- | IL_2 = x\right] \\ &= h_2 x + \underline{g}_1(x|\mathbf{S}),\end{aligned}$$

and so on.

In general, for $j = 1, \ldots, N$, given $\underline{g}_{j-1}$, we have:

$$\hat{g}_j(x|\mathbf{S}) = h_j x + \underline{g}_{j-1}(x|\mathbf{S}) \tag{6.21}$$

$$g_j(y|\mathbf{S}) = \mathbb{E}\left[\hat{g}_j(y - D_j|\mathbf{S})\right] \tag{6.22}$$

$$\underline{g}_j(x|\mathbf{S}) = g_j(\min\{S_j, x\}|\mathbf{S}). \tag{6.23}$$

So for any base-stock vector $\mathbf{S}$ and any known value of IL_j, ITP_j, or IL_{j+1}, we can calculate the expected cost in stages $1, \ldots, j$. What's more, we know ITP_j for $j = N$—it equals S_N. Therefore, the expected cost of the entire system, $g(\mathbf{S})$, is given by $g_N(S_N|\mathbf{S})$.

This gives us a way to calculate the expected cost recursively for a *given* $\mathbf{S}$. We are only a short leap from finding the *optimal* $\mathbf{S}$. Since the recursion for stages $1, \ldots, j-1$ does not depend on S_j, we don't need to choose S_j until we reach $\underline{g}_j(\cdot)$ in the recursion. At that point, we can simply set S_j to the y that minimizes $g_j(\cdot)$. This idea is made concrete in the next theorem. Note that the functions in the theorem omit "$|\mathbf{S}$" since we are choosing $\mathbf{S}$ rather than evaluating the cost for a given $\mathbf{S}$.

Theorem 6.3 *Let $\underline{g}_0(x) = (p + h'_1)x^-$. For $j = 1, \ldots, N$, let*

$$\hat{g}_j(x) = h_j x + \underline{g}_{j-1}(x) \tag{6.24}$$

$$g_j(y) = \mathbb{E}\left[\hat{g}_j(y - D_j)\right] \tag{6.25}$$

$$S_j^* = \text{argmin}\{g_j(y)\} \tag{6.26}$$

$$\underline{g}_j(x) = g_j(\min\{S_j^*, x\}). \tag{6.27}$$

Then $\mathbf{S}^* = (S_j^*)_{j=1}^N$ is the optimal base-stock vector and $g_N(S_N^*)$ is the corresponding optimal cost.

Theorem 6.3 is the result of the groundwork laid by Clark and Scarf (1960) and subsequent refinements by Chen and Zheng (1994). It says that, rather than simultaneously optimizing all of the base-stock levels, we can optimize them sequentially, beginning with stage 1 and working upstream, one stage at a time. Moreover, $g_j(y)$ is known to be convex, so at each iteration we only need to minimize a single-variable, convex function. This theorem underlies much of the theory of multiechelon stochastic-service models. (Zipkin (2000) even goes so far as to call (6.24)–(6.27) the "fundamental equation[s] of supply-chain theory.")

The arguments above imply that, to evaluate the cost of a given (not necessarily optimal) base-stock vector $\mathbf{S}$, we simply skip the optimization step (6.26) and evaluate the functions using $\mathbf{S}$ instead of $\mathbf{S}^*$.

Consider the optimization problem at stage 1. We have:

$$\hat{g}_1(x) = h_1 x + (p + h'_1)x^- \tag{6.28}$$

$$\begin{aligned} g_1(y) &= \mathbb{E}\left[\hat{g}_1(y - D_1)\right] \\ &= \mathbb{E}\left[h_1(y - D_1) + (p + h'_1)(y - D_1)^-\right] \\ &= \mathbb{E}\left[h_1[(y - D_1)^+ - (y - D_1)^-] + (p + h'_1)(y - D_1)^-\right] \\ &= \mathbb{E}\left[h_1(y - D_1)^+ + (p + h'_1 - h_1)(y - D_1)^-\right] \\ &= \mathbb{E}\left[h_1(y - D_1)^+ + (p + h'_2)(D_1 - y)^+\right] \end{aligned} \tag{6.29}$$

This function is identical in form to the newsvendor objective function (4.3), with p replaced by $p + h'_2$. Therefore, from (4.17), $g_1(y)$ is minimized by

$$S_1^* = F_1^{-1}\left(\frac{p + h'_2}{h_1 + p + h'_2}\right) = F_1^{-1}\left(\frac{p + \sum_{i=2}^N h_i}{p + \sum_{i=1}^N h_i}\right). \tag{6.30}$$

At upstream stages, the functions $g_j(y)$ become more complicated and cannot be minimized in closed form. In fact, the expectation in $g_j(y)$ must be evaluated numerically for every candidate value y. Therefore, although (6.26) is a convex minimization problem, it is somewhat computationally expensive to execute, as well as cumbersome to implement.

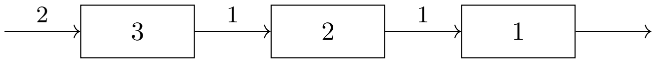


Figure 6.4 3-Stage serial system for Example 6.1.

The function $\hat{g}_j(x)$ is similar to a deterministic cost function, analogous to (4.1) or (5.3)—if we know that $IL_j = x$ at a given time, then the cost rate at that time for stages $1, \ldots, j$ is $\hat{g}_j(x)$. For $j = 1$, (6.28) shows that the form of $\hat{g}_j(x)$ is exactly the same as that of (4.1) or (5.3) since

$$h_1 x + (p + h'_1)x^- = h_1 x^+ + (p + h'_2)x^-.$$

For $j > 1$, $\underline{g}_j(x)$ replaces the stockout penalty term. In fact, $\underline{g}_j(x)$ is sometimes called the *implicit penalty function*. It captures the downstream implications of upstream stockouts.

□ **EXAMPLE 6.1**

Consider the 3-stage serial system in Figure 6.4. Demand at stage 1 is distributed as $N(5, 1^2)$ per unit time. The lead times are $L_1 = L_2 = 1$ and $L_3 = 2$. Local holding costs are given by $(h'_1, h'_2, h'_3) = (7, 4, 2)$, so that the echelon holding costs are $(h_1, h_2, h_3) = (3, 2, 2)$. The stockout cost at stage 1 is $p = 37.12$ per unit time. We will use Theorem 6.3 to find the optimal echelon base-stock levels and the corresponding expected cost.

First, we have $\underline{g}_0(x) = (37.12 + 7)x^-$, $\hat{g}_1(x) = 3x + (37.12 + 7)x^-$, and

$$g_1(y) = \mathbb{E}[\hat{g}_1(y - D_1)] = \mathbb{E}[3(y - D_1)^+ + (37.12 + 4)(D_1 - y)^+]$$

from (6.29). We can solve this numerically, but (6.30) gives us an analytical solution:

$$S_1^* = F_1^{-1}\left(\frac{37.12 + 4}{37.12 + 7}\right) = 6.49,$$

with $g_1(S_1^*) = 5.79$. Figure 6.5(b) plots $\underline{g}_0(x)$, $g_1(y)$, and S_1^*.

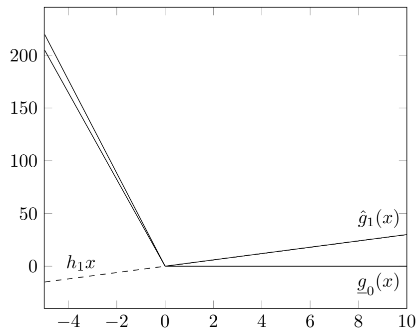
From $g_1(y)$ and S_1^*, we get $\underline{g}_1(x)$ (Figure 6.5(c)), then $\hat{g}_2(x)$ (Figure 6.5(d)), and then $g_2(y)$ (Figure 6.5(e)). Optimizing numerically, we get $S_2^* = 12.02$ and $C_2(S_2^*) = 20.82$.

Continuing in this way, we get $\underline{g}_2(x)$ (Figure 6.5(f)), $\hat{g}_3(x)$ (Figure 6.5(g)), and $g_3(y)$ (Figure 6.5(h)). This function is optimized by $S_3^* = 22.71$, with expected cost $C_3(S_3^*) = 47.65$. By Theorem 6.3, the optimal echelon base-stock levels for this system are $\mathbf{S}^* = (6.49, 12.02, 22.71)$, with optimal expected cost 47.65.

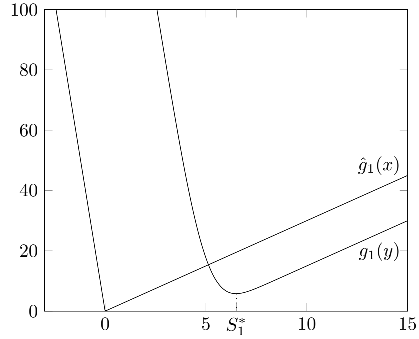
□

6.2.3 Heuristic Approach for Serial Systems

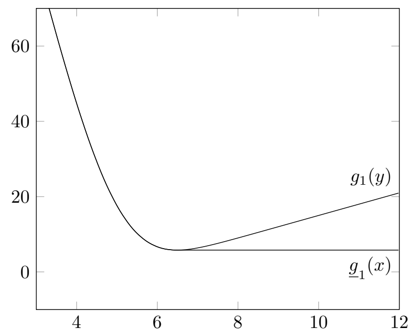
Suppose we have found $S_1^*, \ldots, S_{j-1}^*$, and we now need to find S_j^*. Theorem 6.3 tells us that S_j^* does not depend on the base-stock levels at stages $j + 1, \ldots, N$, although it does indirectly depend on the echelon holding costs at those stages (because $g_j(y)$ includes h'_1). Suppose we truncate the system at stage j (i.e., remove all stages upstream from j), leave



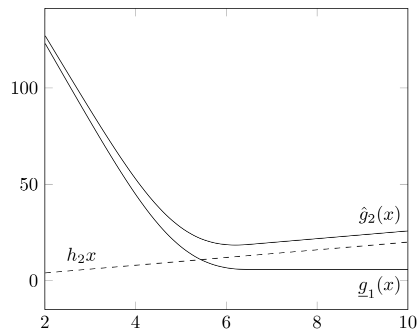
(a) $\underline{g}_0 \to \hat{g}_1$.



(b) $\hat{g}_1 \to g_1$.



(c) $g_1 \to \underline{g}_1$.



(d) $\underline{g}_1 \to \hat{g}_2$.

Figure 6.5 Functions from Theorem 6.3 for Example 6.1.

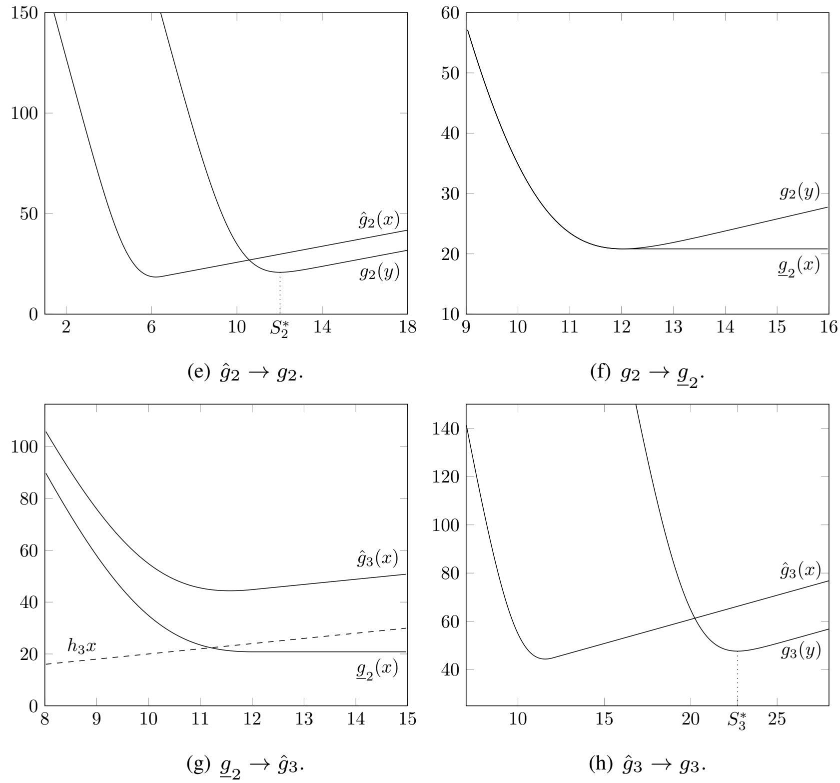


Figure 6.5 Functions from Theorem 6.3 for Example 6.1 (cont'd).

the echelon holding costs at the remaining stages intact, and replace p with $p + h'_{j+1}$. Then the S_j^* that is optimal for stage j in this truncated system is also optimal for stage j in the original system (Shang and Song 2003). In other words, the y that minimizes

$$g_j(y) = \mathbb{E}\left[\sum_{i=1}^{j}\left(\sum_{k=i}^{j} h_k\right)(I'_i(y) + IT_{i-1}(y)) + \left(p + h'_{j+1}\right)B'_1(y)\right] \tag{6.31}$$

also minimizes $g_j(y)$ in (6.26). (In (6.31) we have emphasized that IL and IT are functions of y, and we have truncated the system at j; otherwise, it is identical to (6.8).) We obtained a similar result for stage 1 in (6.29).

Why is this true? Well, in the truncated system, each unit sold reduces the holding cost by $\sum_{i=1}^{j} h_i$, but the true cost reduction, for the original system, is $\sum_{i=1}^{N} h_i = h'_1$. Therefore, there is an extra $h'_1 - \sum_{i=1}^{j} h_i = h'_{j+1}$ in "perceived benefit" for each sale that is not reflected in the holding costs of the truncated system. Similarly, each demand that cannot be satisfied immediately increases the cost by this amount. We therefore model this by adding the perceived benefit, h'_{j+1}, to the original stockout cost, p.

Now, (6.31) is no easier to solve than (6.26)—except for one special case. Suppose that $h'_1 = \cdots = h'_j = h'$, for some fixed h'. (Or, equivalently, $h_1 = \cdots = h_{j-1} = 0$ and $h_j = h'$.) Then it is optimal to hold all of the inventory at stage 1, because upstream inventory is not cheaper, and it requires a longer lead time to reach the customer. We can therefore replace this j-stage system with a single-stage system with a holding cost of h', a stockout cost of $p + h'_{j+1}$, and a lead time of $\sum_{i=1}^{j} L_i$.

This would make the problem easy to solve, but would the solution help us? It turns out that, if we choose good values for h', the resulting cost functions provide bounds on the actual cost function, and the resulting base-stock levels provide bounds on the optimal base-stock levels. Moreover, these bounds can be used to compute heuristic values for S_j^*, which turn out to be remarkably accurate. This approximation was proposed by Shang and Song (2003).

We consider two different values for h'. Let $g_j^l(y)$ be the cost function (6.31) with h'_i replaced by h_j for all i, and let $g_j^u(y)$ be the same function with h'_i replaced by $\sum_{k=1}^{j} h_k$ for all i. Let $\tilde{D}_j$ be the lead-time demand for a single-stage system with lead time $\sum_{i=1}^{j} L_i$, i.e.,

$$\tilde{D}_j = \sum_{i=1}^{j} D_i,$$

and let $\tilde{F}_j(\cdot)$ be its cdf.

Then the functions $g_j^l(y)$ and $g_j^u(y)$ are minimized by

$$S_j^u = \tilde{F}_j^{-1}\left(\frac{p + \sum_{i=j+1}^{N} h_i}{h_j + p + \sum_{i=j+1}^{N} h_i}\right) = \tilde{F}_j^{-1}\left(\frac{p + \sum_{i=j+1}^{N} h_i}{p + \sum_{i=j}^{N} h_i}\right)$$

and

$$S_j^l = \tilde{F}_j^{-1}\left(\frac{p + \sum_{i=j+1}^{N} h_i}{\sum_{k=1}^{j} h_k + p + \sum_{i=j+1}^{N} h_i}\right) = \tilde{F}_j^{-1}\left(\frac{p + \sum_{i=j+1}^{N} h_i}{p + \sum_{i=1}^{N} h_i}\right),$$

respectively.

Theorem 6.4 (Shang and Song (2003)) *For any j and y:*

(a) $g_j^l(y) \leq g_j(y) \leq g_j^u(y)$

(b) $S_j^l \leq S_j^* \leq S_j^u$

The theorem suggests that we can approximate S_j^*, for each j, using a weighted average of S_j^l and S_j^u. In fact, Shang and Song (2003) suggest using a simple average, that is,

$$\tilde{S}_j = \frac{1}{2}\left[\tilde{F}_j^{-1}\left(\frac{p+\sum_{i=j+1}^N h_i}{p+\sum_{i=j}^N h_i}\right) + \tilde{F}_j^{-1}\left(\frac{p+\sum_{i=j+1}^N h_i}{p+\sum_{i=1}^N h_i}\right)\right]. \tag{6.32}$$

If local base-stock levels are desired, we can compute $\tilde{S}'_j$ from $\tilde{S}_j$ as described in Section 6.2.1.

This approximation performs quite well: Shang and Song (2003) report an average error of 0.24% and a maximum error of less than 1.5% on their test instances, where the errors are computed by comparing the heuristic solutions with the exact solutions from Theorem 6.3.

This heuristic can be used for periodic-review systems as well. However, in this case, the lead-times must each be inflated by one unit, assuming the system uses the sequence of events on page 90. See Shang and Song (2003) for details.

☐ **EXAMPLE 6.2**

Return to the serial system in Example 6.1. We will use the Shang–Song heuristic to find approximate values for $\mathbf{S}^*$.

Recall that $\tilde{D}_j$ is the lead-time demand for a single-stage system with lead time $\sum_{i=1}^j L_i$; then:

$$\begin{aligned}
\tilde{D}_1 &\sim N(5\cdot 1, 1^2\cdot 1) = N(5,1)\\
\tilde{D}_2 &\sim N(5\cdot 2, 1^2\cdot 2) = N(10,2)\\
\tilde{D}_3 &\sim N(5\cdot 4, 1^2\cdot 4) = N(20,4).
\end{aligned}$$

We have $(h_1, h_2, h_3) = (3, 2, 2)$. Therefore:

$$\begin{aligned}
S_1^u &= \tilde{F}_1^{-1}\left(\frac{37.12+4}{37.12+7}\right) = 6.49 &\qquad S_1^l &= \tilde{F}_1^{-1}\left(\frac{37.12+4}{37.12+7}\right) = 6.49\\
S_2^u &= \tilde{F}_2^{-1}\left(\frac{37.12+2}{37.12+4}\right) = 12.35 &\qquad S_2^l &= \tilde{F}_2^{-1}\left(\frac{37.12+2}{37.12+7}\right) = 11.71\\
S_3^u &= \tilde{F}_3^{-1}\left(\frac{37.12+0}{37.12+2}\right) = 23.27 &\qquad S_3^l &= \tilde{F}_3^{-1}\left(\frac{37.12+0}{37.12+7}\right) = 22.00.
\end{aligned}$$

Using (6.32), we have

$$\begin{aligned}
\tilde{S}_1 &= \tfrac{1}{2}(6.49+6.49) = 6.49\\
\tilde{S}_2 &= \tfrac{1}{2}(12.35+11.71) = 12.03\\
\tilde{S}_3 &= \tfrac{1}{2}(23.27+22.00) = 22.63.
\end{aligned}$$

These values are very close to $\mathbf{S}^*$ given in Example 6.1, and indeed their costs are very similar: $g(\tilde{\mathbf{S}}) = 47.66$, compared to $g(\mathbf{S}^*) = 47.65$.

☐

6.2.4 Other Network Topologies

Assembly Systems: Assembly systems turn out to be easy to solve—or, at least, no harder than serial systems. Rosling (1989) proves that every assembly system can be transformed to an equivalent serial system. That serial system can be solved using any method available for such systems (for example, the exact method in Section 6.2.2 or the heuristic one in Section 6.2.3). The resulting solution can then be transformed back to a solution for the assembly system. The equivalence between the two systems is exact, meaning that if we solve the serial system optimally, then the transformed solution will be optimal for the assembly system.

Distribution Systems: Unfortunately, distribution systems are much more difficult. In part, the difficulty stems from the fact that, if a given stage has insufficient inventory to meet the orders placed by its successors, it must decide how to allocate the inventory that it does have among them. For example, it may assign items first-come, first-served, or randomly, or based on some priority system. Therefore, in addition to choosing a replenishment policy at each node, we must also choose an allocation policy. Under stochastic demands, even the optimal form of these policies is unknown, let alone the optimal parameters for the policies. Usually, we simply choose a plausible ordering policy (e.g., a base-stock policy) and a plausible allocation policy (e.g., a first-come, first-served policy) and then optimize the parameters under those assumptions.

The simplest type of distribution system is the *one-warehouse, multiple-retailer* (OWMR) system, a two-echelon system with one upstream stage (the "warehouse") and several downstream stages (the "retailers"). The best known exact algorithm for OWMR systems is the *projection algorithm* (Graves 1985, Axsäter 1990), which involves iterating over the possible values for S_0 (the warehouse base-stock level). For each possible value of S_0, we can find the corresponding optimal S_j for the retailers by solving a single-variable, convex optimization problem for each j. However, the total cost is not a convex function of S_0, which means that we must perform an exhaustive search to find S_0^*. Moreover, each evaluation of the objective function requires numerical convolution, a computationally costly calculation.

Several heuristics have been proposed for OWMR and more general distribution systems. Sherbrooke (1968) proposed the so-called "METRIC" model; his method approximates the stochastic lead times generated by the warehouse for the retailers by replacing them with their means. Graves (1985) proposes a two-moment approximation in which a messy distribution necessary to evaluate the cost is replaced by a simpler distribution with the same mean and variance. This approach can also be used to approximate serial systems. Gallego et al. (2007) propose the "restriction–decomposition" heuristic, which involves solving three subheuristics, each of which makes some simplifying assumption to render the model tractable, and then taking the best of the three resulting solutions. Özer and Xiong (2008) propose a heuristic in which the distribution system is decomposed into multiple serial systems, each of which is solved independently, and then the solutions from the serial systems are summed to obtain a solution for the distribution system. A similar approach is used in the "decomposition–aggregation" heuristic by Rong et al. (2017a), which uses a procedure they call "backorder matching" to convert the base-stock levels from the serial system into those for the distribution system. They also propose a more accurate, but more

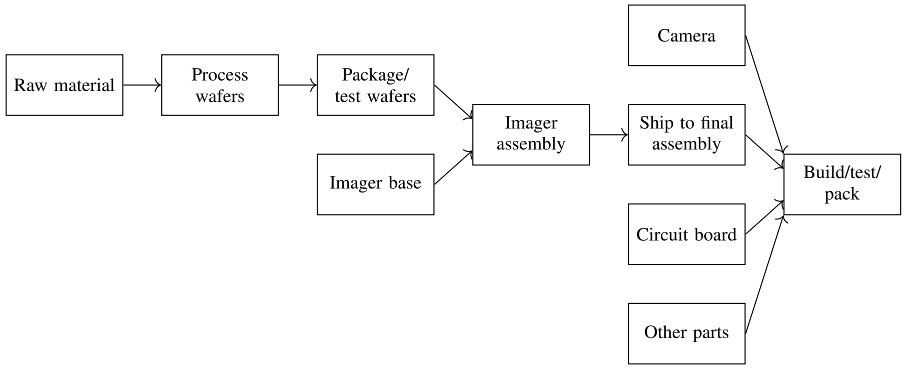


Figure 6.6 Digital camera supply chain network. Reprinted by permission, Graves and Willems, Optimizing strategic safety stock placement in supply chains, *Manufacturing and Service Operations Management*, 2(1), 2000, 68–83. ©2000, the Institute for Operations Research and the Management Sciences (INFORMS), 7240 Parkway Drive, Suite 300, Hanover, MD 21076 USA.

computationally intensive, procedure, called the "recursive optimization" heuristic, which is inspired by Theorem 6.3.

Tree and General Systems: Given the difficulty of solving distribution systems, these more general systems have received little attention in the literature. See, for example, de Kok and Visschers (1999) and de Kok and Fransoo (2003).

6.3 GUARANTEED-SERVICE MODELS

6.3.1 Introduction

Figure 6.6 depicts the supply chain for a digital camera made by Kodak. Each stage represents an activity (as in interpretation (2) from Section 6.1): either a processing activity such as packaging or testing, or an assembly activity such as combining a wafer and an "imager base" to construct an "imager assembly." These activities may occur at different locations or together at the same location. Each stage functions as an autonomous unit that can hold safety stock, place orders to upstream stages, and so on.

The question of interest here is, which stages should hold safety stock, and how much? It may not be necessary for all stages to hold safety stock, but only a few. These stages serve as buffers to absorb all of the demand uncertainty in the supply chain. This problem is a strategic one, since the location of safety stock is a design problem that is costly to change frequently. This problem is therefore known as the *strategic safety stock placement problem* (SSSPP).

The supply chain operates in an infinite-horizon, periodic-review setting, and each stage follows a base-stock policy. Each stage quotes a lead time, or *committed service time* (CST), to its downstream stage(s) within which it promises to deliver each order. As we will see, there is a direct relationship between the CST and the safety stock (and base-stock level) required at each stage. The goal of the strategic safety stock placement model is to choose the CST (and, therefore, the safety stock and base-stock level) at each stage in order to minimize the expected holding cost in each period.

Each stage is required to provide 100% service to its downstream stage(s). In other words, each stage is obligated to deliver every order within the CST *regardless of the size of the order*. In order to enforce this restriction, we will have to assume that the demand is bounded. We will discuss this assumption further in Section 6.3.2.

The guaranteed-service assumption was first used by Kimball in 1955 (later reprinted as Kimball 1988). Simpson (1958) applied it to serial systems and Graves (1988) discussed how to solve the resulting safety stock optimization problem. Inderfurth (1991), Minner (1997), and Inderfurth and Minner (1998) discuss dynamic programming (DP) approaches for distribution and assembly systems. Graves and Willems (2000) extend this to tree systems, and Magnanti et al. (2006) and Humair and Willems (2011) allow general networks that include (undirected) cycles.

We will build gradually to tree networks similar to the one pictured in Figure 6.6, considering first the single-stage case, then serial systems, and finally tree networks. First, we will discuss the demand process.

Throughout Section 6.3, h_i will be used to represent the *local* holding cost at stage i. (In Section 6.2, it represented the echelon holding cost.)

6.3.2 Demand

We assume that the demand *in any interval of time* is bounded. In practice, this is not a terribly realistic assumption (unless the bound is very large), but it is necessary in this model to guarantee 100% service. One way to model the demand is simply to truncate the right tail of the demand distribution. That is, if demand is normally distributed, we simply ignore any demands greater than, say, z_α standard deviations above the mean, for some constant α. This is the approach we will take throughout.

In particular, consider a stage that faces external demand (as opposed to serving other downstream stages). Suppose the demand per period is distributed $N(\mu, \sigma^2)$. Then we will assume that the total demand in any τ periods is bounded by

$$D(\tau) = \mu\tau + z_\alpha\sigma\sqrt{\tau} \tag{6.33}$$

for some constant α. In other words, we assume that the demand in τ consecutive periods is no more than z_α standard deviations above its mean, since the mean demand in τ periods is $\mu\tau$ and the standard deviation is $\sigma\sqrt{\tau}$. This implies that the demand in a single period is bounded by $\mu + z_\alpha\sigma$. The reverse implication, however, is not true: Assuming the single-period demand is bounded by $\mu + z_\alpha\sigma$ implies that the τ-period demand is bounded by $\mu\tau + z_\alpha\sigma\tau$; it does not imply the stronger bound of $\mu\tau + z_\alpha\sigma\sqrt{\tau}$.

If, in actuality, the demand in a given τ-period interval exceeds $D(\tau)$, the excess demands are assumed to be handled in some other manner—say, by outsourcing, scheduling overtime shifts, or by some other method not captured in the model. This allows us to ignore the demands in the tail and pretend the demand never exceeds its bound.

We will use the demand bound in (6.33), but any other bound $D(\tau)$ is acceptable, with suitable changes to the derivations below.

6.3.3 Single-Stage Network

Consider a single stage that quotes a CST of S periods to an external customer. (Recall that S denotes a CST in this section, but a base-stock level in Section 6.2.) The stage receives raw materials from an external supplier, which promises an inbound CST of SI periods. Finally, the stage itself requires a processing time of T periods to perform its function. Items that have been ordered from the supplier but not yet received are referred to as *on-order* inventory; those that have been received from the supplier and are currently being processed are referred to as *work-in-progress* (WIP) inventory; and those that have completed their processing are referred to as *finished-goods* inventory. (See Figure 6.7.) SI and T are both constants (parameters). S is the decision variable. Our goal in this section is to determine the amount of safety stock required if the stage quotes a CST of S periods.

The inventory position equals the finished-goods inventory, plus the on-order and WIP inventory, minus demands that have occurred but have not yet been satisfied. These unmet demands would be considered backorders in the stochastic-service model, but in the guaranteed-service model, they are acceptable as long as they are satisfied within S periods. Thus, they are subtracted from the inventory position just as backorders are, but they are not penalized in the objective function.

The sequence of events in period t in the guaranteed-service model is as follows:

1. The inventory position, IP_t, is calculated.
2. The demand, d_t, is observed.
3. A replenishment order of size $y - (IP_t - d_t)$ is placed, where y is the base-stock level.
4. Items that were ordered from the supplier SI periods ago are added to WIP inventory.
5. Items that entered WIP inventory T periods ago are added to finished-goods inventory.
6. Items that were demanded S periods ago are removed from finished-goods inventory.
7. Holding costs are assessed based on the ending inventory level.

Note that this sequence of events assumes that the demand is observed *before* the order is placed, whereas the stochastic-service, periodic-review models in Chapter 4 assume the demand is observed *after*. Actually, the two are mathematically equivalent since we can simply add 1 to a stochastic-service lead time and then apply the guaranteed-service sequence of events, or subtract 1 from a guaranteed-service lead time and then apply the stochastic-service sequence of events.[3]

Other differences between the sequences of events in the stochastic- and guaranteed-service models are more cosmetic. For example, the guaranteed-service sequence includes WIP inventory in the inventory position and subtracts items that have been demanded but not yet satisfied. One can consider the same as happening in the stochastic-service model, in which both of these quantities equal 0.

S is similar to a "demand lead time"—i.e., an advance warning of demands that must be met in the future. Conversely, SI and T both contribute to the supply lead time, since $SI + T$ periods elapse between when the stage places an order and when the products are ready to be delivered to the stage's customer. Each unit increase in demand lead time

[3]If $L = 0$ in the guaranteed-service model, this means we use a lead time of -1 in the stochastic-service model. This doesn't make sense for actual lead times but it is acceptable mathematically.

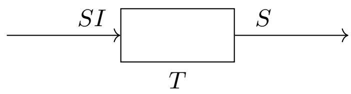


Figure 6.7 Single-stage network.

is equivalent to a unit decrease in the supply lead time. (This claim should make sense intuitively; see Hariharan and Zipkin (1995) for a rigorous proof in a somewhat different context.) Therefore, this system is equivalent to a system with no demand lead time and with $SI + T - S$ periods of supply lead time. The quantity $SI + T - S$ is called the *net lead time* (NLT).

The local base-stock level required at the stage is equal to the demand bound:

$$y = \mu(SI + T - S) + z_\alpha \sigma \sqrt{SI + T - S}. \tag{6.34}$$

(If the demand bound $D(\tau)$ takes a form other than that given in (6.33), we simply replace the right-hand side with the appropriate bound.) This expression is analogous to (4.46), with the net lead time $SI + T - S$ replacing the lead time L. (The "+1" in (4.46) does not appear here because of the difference in the sequence of events, discussed above.)

If the base-stock level is set according to (6.34), then the stage will always be able to meet any demand within S periods. This is a result of the conservation of flow argument that we made in Section 4.3.4.1: In period t, we place an order to bring the inventory position up to the base-stock level, y. By period $t + (SI + T)$, all of these units will have arrived, been processed, and been added to inventory. In other words, if no additional demands occur between period $t + 1$ and $t + (SI + T)$, the on-hand inventory at the end of period $t + (SI + T)$ will equal y. This quantity needs to be sufficient to meet all demands that are due before period $t + (SI + T)$, in other words, demands occurring between period $t + 1$ and $t + (SI + T - S)$. The demand in these $SI + T - S$ periods will be no more than $\mu(SI + T - S) + z_\alpha \sigma \sqrt{SI + T - S}$, so we should set y equal to this value, as in (6.34).

Note that this argument ignores the units that were demanded during periods $t - S + 1$ through t. These demands also must be satisfied out of the items that are on-order at time t. But these items are subtracted from the inventory position in period t. Therefore, the on-order items include items to meet these demands, in addition to the y items that are available in period $t + (SI + T)$.

Given the base-stock level in (6.34), the safety stock is *approximately* equal to

$$z_\alpha \sigma \sqrt{SI + T - S} \tag{6.35}$$

(since base stock = cycle stock + safety stock). The reason this expression is only approximate lies in the way we truncate the normal distribution. We have truncated the distribution z_α standard deviations above the mean, and at 0. The truncation is therefore not symmetric, and so the mean of the revised distribution no longer equals μ. Therefore, the mean demand over the NLT is not exactly equal to $\mu(SI + T - S)$, so the safety stock is not exactly equal to the expression given in (6.35). (The true safety stock level is greater.) As z_α increases, the approximation improves. To take an extreme example, if $\alpha < 0.5$, then $z_\alpha < 0$, so the approximate safety stock is negative, even though the true safety stock may not be. The same situation can also cause the expected holding cost per period, given below, to be negative. Therefore, in what follows, we will require $\alpha > 0.5$ and will assume that it is large enough that we can treat (6.35) as though it were exact.

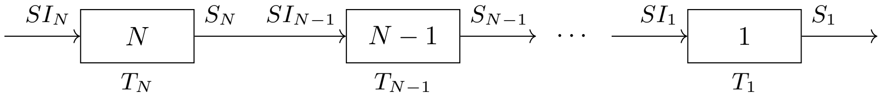


Figure 6.8 N-stage serial system in guaranteed-service model.

From (6.35), as the CST increases, the safety stock level decreases. At one extreme, the stage can quote a CST of $S = SI + T$, in which case every time the stage receives an order, it can place an order to its supplier, wait for it to arrive, process it, and deliver it in time—it has to hold 0 safety stock since $\sqrt{SI + T - (SI + T)} = 0$. At the other extreme, the stage can quote a CST of $S = 0$, in which case delivery is required immediately, so the stage must hold the maximum possible safety stock: $z_\alpha \sigma \sqrt{SI + T}$. Or the stage can quote some CST strictly between 0 and $SI + T$ and hold safety stock strictly between $z_\alpha \sigma \sqrt{SI + T}$ and 0.

If the holding cost is h per unit per time period (charged on ending inventory, as usual), then the expected holding cost per period is

$$h z_\alpha \sigma \sqrt{SI + T - S} \tag{6.36}$$

since the expected ending inventory is equal to the safety stock. From now on, we will focus on the safety stock level rather than the base-stock level since optimizing one is equivalent to optimizing the other.

6.3.4 Serial Systems

Now consider a serial supply chain network such as the one pictured in Figure 6.8. Each stage follows the same sequence of events as in Section 6.3.3. The notation from that section will now include subscripts i to refer to a given stage. Note that $SI_{N-1} = S_N$ (stage $N-1$'s inbound time is equal to stage N's outbound time), $SI_{N-2} = S_{N-1}$, and so on. And stage N's inbound time is from an external supplier rather than from another stage.

The expected holding cost per period is

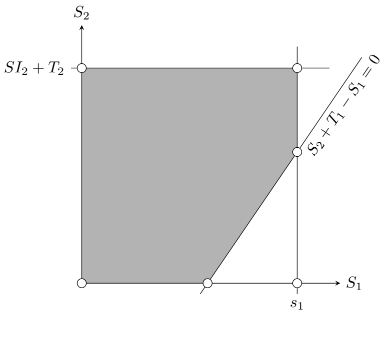
$$g(\mathbf{S}) = \sum_{i=1}^{N} h_i z_\alpha \sigma \sqrt{SI_i + T_i - S_i}, \tag{6.37}$$

where $\mathbf{S} = (S_1, \ldots, S_N)$ and h_i is the local holding cost at stage i. Note that the same σ is used at all stages, since each stage places an order equal to the order that it received.

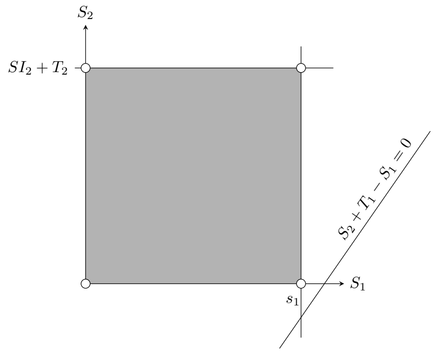
Obviously, with no constraints on the CST to the external customer (downstream from stage 1), the optimal solution would be to set $S_i = SI_i + T_i$ for all i; this solution has 0 holding cost because no safety stock is held. Therefore, we will assume that the CST to the external customer is already set to some constant s_1, and we require $S_1 \le s_1$. But it will never be to our advantage to set $S_1 < s_1$, so in general, we can assume $S_1 = s_1$. Only $S_2, \ldots, S_N$, then, are really decision variables.

For each $i = 2, \ldots, N$, g is concave in S_i since

$$\frac{\partial g}{\partial S_i} = -\frac{1}{2} h_i z_\alpha \sigma (SI_i + T_i - S_i)^{-\frac{1}{2}} + \frac{1}{2} h_{i-1} z_\alpha \sigma (S_i + T_{i-1} - S_{i-1})^{-\frac{1}{2}}$$



(a) $s_1 - T_1 \geq 0$.



(b) $s_1 - T_1 < 0$.

Figure 6.9 Feasible region for two-stage system.

$$\frac{\partial^2 g}{\partial S_i^2} = -\frac{1}{4} h_i z_\alpha \sigma (SI_i + T_i - S_i)^{-\frac{3}{2}} - \frac{1}{4} h_{i-1} z_\alpha \sigma (S_i + T_{i-1} - S_{i-1})^{-\frac{3}{2}} < 0.$$

Therefore, the optimal solution occurs at the extreme points—each S_i is set to its minimum or maximum feasible value. What are the minimum and maximum? Well, $S_i \leq SI_i + T_i$, otherwise the quantity under the square root for i in (6.37) is negative. Similarly, $S_i \geq S_{i-1} - T_{i-1}$, otherwise the quantity under the square root for $i-1$ is negative. But we also know that $S_i \geq 0$. Therefore, the limits of S_i are $\max\{0, S_{i-1} - T_{i-1}\}$ and $SI_i + T_i$; the optimal solution has S_i^* taking on one of these two values.

To illustrate this graphically, suppose $N = 2$. In effect, we are trying to solve the following IP:

$$\begin{aligned} \text{minimize} \quad & h_2 z_\alpha \sigma \sqrt{SI_2 + T_2 - S_2} + h_1 z_\alpha \sigma \sqrt{S_2 + T_1 - S_1} && (6.38)\\ \text{subject to} \quad & SI_2 + T_2 - S_2 \geq 0 && (6.39)\\ & S_2 + T_1 - S_1 \geq 0 && (6.40)\\ & S_1 \leq s_1 && (6.41)\\ & S_1, S_2 \geq 0 \text{ and integer} && (6.42) \end{aligned}$$

The feasible region for this IP is pictured in Figure 6.9; part (a) assumes that $s_1 - T_1 \geq 0$ while part (b) assumes that $s_1 - T_1 < 0$. If we assume that $S_1 = s_1$, then only the right-hand edge of the feasible region is relevant; the extreme points on this edge are $S = (s_1, SI_2 + T_2)$ and $S = (s_1, s_1 - T_1)$, as expected.

This logic can be used to prove the following:

Theorem 6.5 *Suppose $s_1 = 0$ (immediate service is required to the customer). Then for all $i = 2, \ldots, N$, either $S_i^* = 0$ or $S_i^* = S_{i+1}^* + T_i$.*

Proof. Omitted; see Problem 6.6. ■

In other words, each stage follows an "all-or-nothing" inventory policy: either it holds 0 safety stock and quotes the maximum possible CST, or it holds the maximum possible safety stock and quotes 0 CST. We will see shortly that this property does not hold for the tree systems considered in Section 6.3.5.

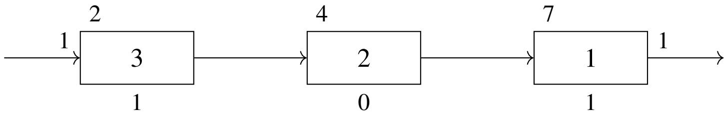


Figure 6.10 Example network for SSSPP DP algorithm for serial systems.

The mathematical program (6.38)–(6.42) is not usually solved directly. Instead, we can solve this problem using DP (see Inderfurth 1991). Let $\theta_k(SI)$ equal the optimal cost in stages $1, \ldots, k$ if stage k receives an inbound CST of SI. Then $\theta_k(SI)$ can be computed recursively as follows:

$$\theta_1(SI) = h_1 z_\alpha \sigma \sqrt{SI + T_1 - s_1} \tag{6.43}$$

$$\theta_k(SI) = \min_{0 \le S \le SI + T_k} \left\{ h_k z_\alpha \sigma \sqrt{SI + T_k - S} + \theta_{k-1}(S) \right\} \tag{6.44}$$

Equation (6.43) initializes the recursion: At stage 1, for any inbound CST SI, the NLT is $SI + T_1 - s_1$ since the outbound CST is fixed at s_1. Then (6.44) calculates $\theta_k(SI)$ recursively: If stage k receives an inbound CST of SI and we choose an outbound CST of S, the cost at stage k is $h_k z_\alpha \sigma \sqrt{SI + T_k - S}$, and the cost at stages $1, \ldots, k-1$ is $\theta_{k-1}(S)$ since stage $k-1$ will receive an inbound CST of S. The right-hand side of (6.44) chooses the S that minimizes this cost, subject to the constraint that $0 \le S \le SI + T_k$ to ensure that S and the NLT are both nonnegative.

The recursive equations (6.43)–(6.44) must be evaluated for each stage k and for each possible SI. We therefore need to determine which values SI can take on at stage k. Clearly, $SI \ge 0$. Furthermore, if, at stage k, $SI > SI_N + \sum_{j=k+1}^{N} T_j$, then the NLT will be negative at some stage. Therefore, at stage k, we can restrict our attention to $0 \le SI \le SI_N + \sum_{j=k+1}^{N} T_j$.

In the next section, we will generalize this approach to solve tree systems.

☐ **EXAMPLE 6.3**

Consider the network pictured in Figure 6.10. The numbers below the stages are the processing times T_i. The number on the inbound arrow to stage 3 indicates that $SI_3 = 1$, while the outbound number from stage 1 indicates that the fixed CST $s_1 = 1$. The holding costs at stages 1, 2, and 3 are 7, 4, and 2, respectively, and are noted above each stage. Assume $z_\alpha = \sigma_i = 1$ at all stages.

First note that $SI_N + \sum_{j=k+1}^{N} T_j = 2$ at stages 1 and 2. (SI is fixed to 1 at stage 3.) These are the maximum SI values that we must consider at each stage.

We consider stage $k = 1$ first. From (6.43), $\theta_1(SI) = 7\sqrt{SI}$ for all SI:

$$\begin{aligned} \theta_1(0) &= 0 \\ \theta_1(1) &= 7 \\ \theta_1(2) &= 7\sqrt{2} = 9.90. \end{aligned}$$

Next, at stage 2, we use (6.44):

$$\theta_2(0) = \min_{S=0}\{4\sqrt{0 + 0 - S} + \theta_1(S)\} = 0$$

$$\begin{aligned}
\theta_2(1) &= \min_{0\le S\le 1}\{4\sqrt{1+0-S}+\theta_1(S)\}\\
&= \min\{4+0, 0+7\} = 4\\
\theta_2(2) &= \min_{0\le S\le 2}\{4\sqrt{2+0-S}+\theta_1(S)\}\\
&= \min\{4\sqrt{2}+0, 4+7, 0+7\sqrt{2}\} = 4\sqrt{2} = 5.6.
\end{aligned}$$

Finally, at stage 3, we have only one SI value to consider since SI_3 is fixed at 1:

$$\begin{aligned}
\theta_3(1) &= \min_{0\le S\le 2}\{2\sqrt{1+1-S}+\theta_2(S)\}\\
&= \min\{2\sqrt{2}+0, 2+4, 0+5.6\}\\
&= 2\sqrt{2} = 2.83.
\end{aligned}$$

Since $S = 0$ solved the minimization for $\theta_3(1)$, we have $S_3^* = 0$. Therefore, $SI = 0$ at stage 2, and $S = 0$ solved the minimization for $\theta_2(0)$ as well. Finally, $s_1 = 1$. Therefore, the optimal CSTs are $\mathbf{S}^* = (0, 0, 1)$. These CSTs imply that the NLTs at stages 1, 2, and 3 are 0, 0, and 2, respectively. Therefore, the optimal safety stock levels are as follows:

$$\begin{aligned}
SS_1 &= \sqrt{0} = 0.00\\
SS_2 &= \sqrt{0} = 0.00\\
SS_3 &= \sqrt{2} = 1.41
\end{aligned}$$

□

6.3.5 Tree Systems

At this point, we will turn our attention to tree systems. The model and algorithm described here were introduced by Graves and Willems (2000). (See also Graves and Willems (2003b) for an erratum.) Their algorithm runs in pseudopolynomial time. Lesnaia (2004) provides a polynomial-time implementation that runs in $O(N^3)$ time, where N is the number of stages in the network. For general systems, which may include (undirected) cycles, the problem is NP-hard (Chu and Shen 2003, Lesnaia 2004). See Magnanti et al. (2006) for a solution method based on integer programming techniques and Humair and Willems (2011) for exact and heuristic algorithms that extend the DP algorithm by Graves and Willems (2000) to general systems. Humair et al. (2013) extend the approach to allow stochastic lead times, and Graves and Schoenmeyr (2016) consider capacity constraints.

Let A be the set of (directed) arcs in the network; then stage i is a predecessor to stage j if and only if $(i, j) \in A$. A *demand stage* is a stage that faces external demand. We assume that a stage is a demand stage if and only if it has no successors. It is possible for a tree network to have more than one demand stage. The CST S_i for any demand stage i is set equal to $s_i \ge 0$, a constant, as in Section 6.3.4. Similarly, stages with no predecessors are called *supply stages*. If i is a supply stage, then i receives product from an external supplier with CST $SI_i \ge 0$. It is possible that a nondemand stage could have an external customer in addition to its successors, or that a nonsupply stage could have an external supplier in addition to its predecessors, but we will rule out this possibility to keep things simpler.

Each demand stage i sees periodic demand distributed as $N(\mu_i, \sigma_i^2)$. Nondemand stages see demand that is derived from the stages they serve, and their safety stock levels must be set using the standard deviation of that demand. The standard deviation of demand at stage i (a nondemand stage) is

$$\sigma_i = \sqrt{\sum_{(i,j)\in A} \sigma_j^2} \tag{6.45}$$

since its variance is the sum of the variances of the downstream demands (derived or actual). The amount of safety stock required at stage i is therefore

$$z_\alpha \sigma_i \sqrt{SI_i + T_i - S_i} \tag{6.46}$$

and the expected holding cost at i is

$$h_i z_\alpha \sigma_i \sqrt{SI_i + T_i - S_i} \tag{6.47}$$

whether i is a supply stage, a demand stage, or neither. (Again, h_i is the local holding cost.)

If stage i has more than one successor, we will assume that it quotes *the same CST to all downstream neighbors*. Now, suppose stage i has more than one predecessor. Stage i cannot begin its processing until *all* of the raw materials have arrived. Therefore, if the upstream neighbors quote different CSTs, the effective inbound time at stage i is the maximum of the CSTs of the upstream neighbors. That is,

$$SI_i = \max_{(j,i)\in A} \{S_j\}. \tag{6.48}$$

All of this will be important in the algorithm we use to solve this problem.

Since the objective function is concave in every S_i, the optimal solution occurs at the extreme points, as in the serial-system case. But the "all-or-nothing" result from Theorem 6.5 does not hold, even if $s_i = 0$ for every demand stage. That is, it is not necessarily true that every stage either quotes 0 CST or holds 0 safety stock. An example is pictured in Figure 6.11. The processing time T_i is listed below each stage, and the holding cost h_i is listed above. The inbound CST at the supply stages 3 and 4 is 0, as is the outbound CST at the demand stages 1 and 2. Stage 4 has a very large holding cost, which means it is optimal to hold no safety stock there; therefore, $S_4^* = 4$. We will show that $S_3^* = 4$ as well, even though this means stage 3 quotes a positive CST *and* holds positive safety stock. First suppose $S_3^* < 4$. Then the safety stock level at 3 increases, but there is no decrease in safety stock at stage 1 since stage 4 quotes an inbound time of 4 and $SI_1 = \max\{S_4, S_3\}$. Now suppose $S_3 > 4$. This increases the safety stock required at stage 1, which is quite expensive; the cost more than offsets any savings in holding cost at stage 3. Therefore, $S_3^* = 4$.

6.3.6 Solution Method

We will solve the SSSPP on a tree system using DP. In principle, the approach is similar to the DP for the serial system in Section 6.3.4, but it is more complicated for two main reasons. First, computing the cost of a given decision is trickier than in the serial system. Second, in the serial system, it is clear which stage follows a given stage, and hence, how the DP recursion should be structured. In this problem, this is less clear, since each stage may have more than one upstream and/or downstream neighbor.

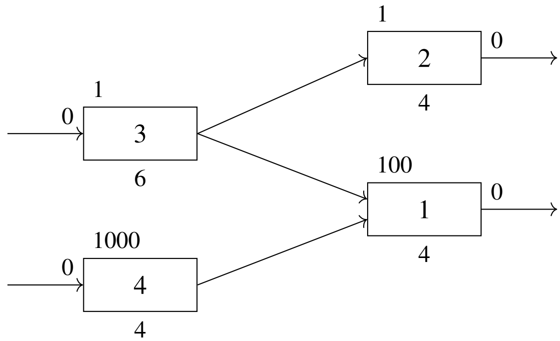
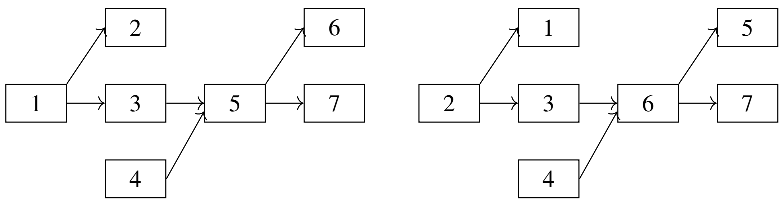


Figure 6.11 A counterexample to the "all-or-nothing" claim for tree systems.



(a) Original network. (b) Relabeled network.

Figure 6.12 Relabeling the network.

6.3.6.1 Labeling the Stages We will address the second issue first. The DP algorithm requires us to relabel the stages so that each stage (other than stage N) *has exactly one adjacent stage with a higher index.* When we describe the algorithm, it will be clear why this is required. The relabeling is performed using Algorithm 6.1. In the algorithm, L represents the set of stages that have been labeled so far and U represents the set of unlabeled stages.

Algorithm 6.1 Relabel stages

```
L ← ∅, U ← {1, ..., N}                                        ▷ Initialization
for k = 1, ..., N do                                          ▷ Labeling stages
    choose i ∈ U such that i is adjacent to at most one other stage in U
    label i with index k
    L ← L ∪ {i}, U ← U \ {i}
end for
return labels
```

☐ **EXAMPLE 6.4**

Consider the network pictured in Figure 6.12(a). Applying the procedure to this network yields the renumbered network in Figure 6.12(b). Note that in this network, every stage has exactly one neighbor (either upstream or downstream) with a higher index, other than stage 7. ☐

6.3.6.2 Functional Equations Next, we describe how to evaluate the cost of a decision at a given stage. We had one recursive function, θ_k, in Section 6.3.4. In this section, we will need two. Each stage will use one function or the other based on whether the DP has already evaluated the stage's successor or its predecessor.

Let M_k be the maximum possible CST at stage k: M_k is equal to the length of the longest path through the network up to stage k, assuming each stage quotes the maximum possible CST of $SI + T$.

For a given stage k, $k = 1, \ldots, N-1$, let p_k be the stage adjacent to k with the higher index in the relabeled network. Also, let N_k be the set of nodes in $\{1, 2, \ldots, k\}$ that are connected (not necessarily adjacent) to k in the undirected subgraph with node set $\{1, 2, \ldots, k\}$. That is,

$$N_k = \{k\} \cup \bigcup_{\substack{(i,k)\in A \\ i<k}} N_i \cup \bigcup_{\substack{(k,j)\in A \\ j<k}} N_j. \tag{6.49}$$

For example, in Figure 6.12(b),

$$\begin{aligned} N_3 &= \{1, 2, 3\} \\ N_4 &= \{4\} \\ N_5 &= \{5\}. \end{aligned}$$

In the course of the DP, decisions made at stage k affect only those stages in N_k. The type of decision made depends on whether p_k is downstream or upstream from k:

- If p_k is *downstream* from k, then the decision to be made is the *outbound* CST S from stage k. The expected holding cost in N_k if k has an outbound CST of S is denoted $\theta_k^o(S)$. (The superscript o stands for "outbound.")
- If p_k is *upstream* from k, then the decision to be made is the *inbound* CST SI to stage k. The expected holding cost in N_k if k has an inbound CST of SI is denoted $\theta_k^i(SI)$. (The superscript i stands for "inbound.")

$\theta_k^o(S)$ and $\theta_k^i(SI)$ are the functional equations for the DP algorithm.

To compute $\theta_k^o(S)$ and $\theta_k^i(SI)$, we first compute the expected holding cost for N_k as a function of both the inbound and outbound CSTs at node k:

$$\begin{aligned} c_k(S, SI) =& h_k z_\alpha \sigma_k \sqrt{SI + T_k - S} \\ &+ \sum_{\substack{(i,k)\in A \\ i<k}} \min_{0 \le x \le SI} \{\theta_i^o(x)\} + \sum_{\substack{(k,j)\in A \\ j<k}} \min_{S \le y \le M_j - T_j} \{\theta_j^i(y)\}. \end{aligned} \tag{6.50}$$

The first term is simply the expected holding cost at node k. The second term is the cost at nodes in N_k that are upstream from k. For a stage i that is immediately upstream from k, if k's inbound CST is SI then i's outbound CST is *at most* SI. Why "at most" instead of "equal to"? Remember that at node k, SI is the maximum of the S's from all upstream neighbors. Forcing S to equal SI for all upstream neighbors is probably not optimal. Similarly, the third term is the cost at nodes in N_k that are downstream from k. For a stage j that is immediately downstream from k, if k's outbound CST is S then j's inbound CST is *at least* S. It's not necessarily equal to S since j might have other upstream neighbors that quote CSTs longer than S.

At stage k in the DP, we know $\theta_i^o(S)$ for $i < k$ and $\theta_j^i(SI)$ for $j < k$ because we have already visited all stages with smaller indices than k. At those stages, we have computed $\theta_i^o(S)$ for all possible values of S and $\theta_j^i(SI)$ for all possible values of SI.

To compute $\theta_k^o(S)$ for a given S, we set

$$\theta_k^o(S) = \min_{SI}\{c_k(S, SI)\}. \tag{6.51}$$

In other words, if we want to set k's outbound CST to S, we determine the cheapest possible inbound CST given that the outbound CST is S. What should the minimum be taken over (that is, what values of SI are legal)? If k is a supply node (no upstream neighbors), then there is only one possible value for SI: SI_k, a constant. But if k is not a supply node, then SI could be anywhere between $\max\{0, S - T_k\}$ (to ensure the quantity under the square root is positive) and $M_k - T_k$, where M_k is as defined above.

Similarly, to compute $\theta_k^i(SI)$ for a given SI, we set

$$\theta_k^i(SI) = \min_{S}\{c_k(S, SI)\}. \tag{6.52}$$

What are the limits of S? If k is a demand stage (no downstream neighbors), then we have to set $S = s_k$. Otherwise, S can be anywhere between 0 and $SI + T_k$.

6.3.6.3 *Dynamic Programming Algorithm*

Algorithm 6.2 gives the pseudocode for the DP algorithm.

Algorithm 6.2 DP algorithm for tree SSSPP

1: **for** $k = 1, \ldots, N-1$ **do**
2:     **if** p_k is downstream from k **then**
3:         calculate $\theta_k^o(S)$ for $S = 0, 1, \ldots, M_k$
4:     **else**
5:         calculate $\theta_k^i(SI)$ for $SI = 0, 1, \ldots, M_k - T_k$
6:     **end if**
7: **end for**
8: $SI^* \leftarrow \text{argmin}_{SI=0,1,\ldots,M_N-T_N}\ \theta_N^i(SI)$
9: **return** $\theta_N^i(SI^*)$

The algorithm returns the optimal objective value, which is equal to the minimum value of $\theta_N^i(SI)$ found in line 8. The optimal solution is found by “backtracking,” similar to the Wagner–Whitin algorithm.

Here’s why the algorithm works. Suppose we’re at stage $k < N$ in step 1. We know that k has exactly one neighbor with higher index, called p_k. If p_k is downstream from k, then we compute the cost of setting k’s outbound CST S to each possible value. Computing the cost for a given value, $\theta_k^o(S)$, requires knowing $c_k(S, SI)$, which in turn requires knowing $\theta_i^o(x)$ for all stages i that are immediately upstream and $\theta_j^i(y)$ for all stages j that are immediately downstream from k, for all appropriate values of x and y. We know that for every upstream i, we computed $\theta_i^o(\cdot)$ in step 1(a), not $\theta_i^i(\cdot)$ in step 1(b), because i’s neighbor with a higher index is k, which is downstream from it. Similarly, for every downstream j, we computed $\theta_j^i(\cdot)$, not $\theta_i^o(\cdot)$, because $p_j = k$ and k is upstream from j. If, instead, p_k is upstream from k, the logic is similar.

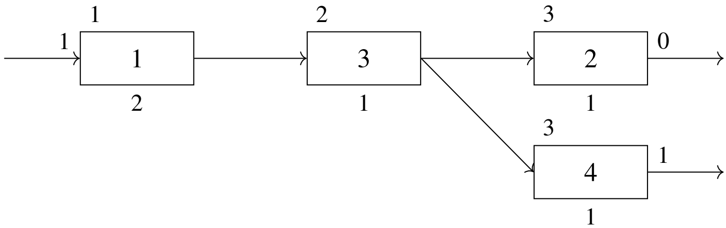


Figure 6.13 Example network for SSSPP DP algorithm for tree systems.

☐ **EXAMPLE 6.5**

We will illustrate the algorithm on the network pictured in Figure 6.13. The numbers below the stages are the processing times T_i. The number on the inbound arrow to stage 1 indicates that $SI_1 = 1$, while the outbound numbers from stages 2 and 4 indicate the fixed CSTs s_i. The holding costs are noted above each stage and are equal to 1 at the first echelon (stage 1), 2 at the second echelon (stage 3), and 3 at the third echelon (stages 2 and 4). Assume $z_\alpha = 1$ at all stages and $\sigma_2 = \sigma_4 = 1$; then $\sigma_1 = \sigma_3 = \sqrt{2}$. Note that the stages have already been relabeled so that each stage has exactly one neighbor with a higher index. Examining the longest path to each node, we get $M_1 = 3$, $M_2 = 5$, $M_3 = 4$, $M_4 = 5$.

Since $p_1 = 3$ is downstream from 1, we first compute $\theta_1^o(S)$ for $S = 0, \ldots, M_1 = 3$. Since 1 is a supply stage, the minimum over SI only considers $SI = 1$.

$$\theta_1^o(0) = \min_{SI=1}\{c_1(0, SI)\} = c_1(0,1) = 1\sqrt{2}\sqrt{1+2-0} = 2.45$$
$$\theta_1^o(1) = \min_{SI=1}\{c_1(1, SI)\} = c_1(1,1) = 1\sqrt{2}\sqrt{1+2-1} = 2.00$$
$$\theta_1^o(2) = \min_{SI=1}\{c_1(2, SI)\} = c_1(2,1) = 1\sqrt{2}\sqrt{1+2-2} = 1.41$$
$$\theta_1^o(3) = \min_{SI=1}\{c_1(3, SI)\} = c_1(3,1) = 1\sqrt{2}\sqrt{1+2-3} = 0.00$$

Next, we compute $\theta_2^i(SI)$ since $p_2 = 3$ is upstream from 2; we need to consider $SI = 0, \ldots, M_2 - T_2 = 4$. Since 2 is a demand stage, the minimum over S only considers $S = 0$.

$$\theta_2^i(0) = \min_{S=0}\{c_2(S, 0)\} = c_2(0,0) = 3\sqrt{0+1-0} = 3.00$$
$$\theta_2^i(1) = \min_{S=0}\{c_2(S, 1)\} = c_2(0,1) = 3\sqrt{1+1-0} = 4.24$$
$$\theta_2^i(2) = \min_{S=0}\{c_2(S, 2)\} = c_2(0,2) = 3\sqrt{2+1-0} = 5.20$$
$$\theta_2^i(3) = \min_{S=0}\{c_2(S, 3)\} = c_2(0,3) = 3\sqrt{3+1-0} = 6.00$$
$$\theta_2^i(4) = \min_{S=0}\{c_2(S, 4)\} = c_2(0,4) = 3\sqrt{4+1-0} = 6.71$$

Now comes the interesting case: stage 3. We need to compute $\theta_3^o(S)$ for $S = 0, \ldots, M_3 = 4$. The minimum over SI ranges from $\max\{0, S - T_3\}$ to $4 - 1 = 3$. Note that $\theta_1^o(x)$ is decreasing in x and $\theta_2^i(y)$ is increasing in y for this network. Therefore, in (6.50),

$$\min_{0 \le x \le SI}\{\theta_1^o(x)\} = \theta_1^o(SI)$$

and

$$\min_{S \le y \le M_2 - T_2} \{\theta_2^i(y)\} = \theta_2^i(S)$$

for all SI and S, and we have:

$$\begin{aligned}
\theta_3^o(0) &= \min_{0 \le SI \le 3} \{c_3(0, SI)\} = 8.28 \\
&\quad c_3(0,0) = 2\sqrt{2}\sqrt{0+1-0} + \theta_1^o(0) + \theta_2^i(0) = 2.83 + 2.45 + 3.00 = 8.28 \\
&\quad c_3(0,1) = 2\sqrt{2}\sqrt{1+1-0} + \theta_1^o(1) + \theta_2^i(0) = 4.00 + 2.00 + 3.00 = 9.00 \\
&\quad c_3(0,2) = 2\sqrt{2}\sqrt{2+1-0} + \theta_1^o(2) + \theta_2^i(0) = 4.90 + 1.41 + 3.00 = 9.31 \\
&\quad c_3(0,3) = 2\sqrt{2}\sqrt{3+1-0} + \theta_1^o(3) + \theta_2^i(0) = 5.66 + 0.00 + 3.00 = 8.66 \\
\theta_3^o(1) &= \min_{0 \le SI \le 3} \{c_3(1, SI)\} = 6.69 \\
&\quad c_3(1,0) = 2\sqrt{2}\sqrt{0+1-1} + \theta_1^o(0) + \theta_2^i(1) = 0.00 + 2.45 + 4.24 = 6.69 \\
&\quad c_3(1,1) = 2\sqrt{2}\sqrt{1+1-1} + \theta_1^o(1) + \theta_2^i(1) = 2.83 + 2.00 + 4.24 = 9.06 \\
&\quad c_3(1,2) = 2\sqrt{2}\sqrt{2+1-1} + \theta_1^o(2) + \theta_2^i(1) = 4.00 + 1.41 + 4.24 = 9.65 \\
&\quad c_3(1,3) = 2\sqrt{2}\sqrt{3+1-1} + \theta_1^o(3) + \theta_2^i(1) = 4.90 + 0.00 + 4.24 = 9.14 \\
\theta_3^o(2) &= \min_{1 \le SI \le 3} \{c_3(2, SI)\} = 7.20 \\
&\quad c_3(2,1) = 2\sqrt{2}\sqrt{1+1-2} + \theta_1^o(1) + \theta_2^i(2) = 0.00 + 2.00 + 5.20 = 7.20 \\
&\quad c_3(2,2) = 2\sqrt{2}\sqrt{2+1-2} + \theta_1^o(2) + \theta_2^i(2) = 2.83 + 1.41 + 5.20 = 9.44 \\
&\quad c_3(2,3) = 2\sqrt{2}\sqrt{3+1-2} + \theta_1^o(3) + \theta_2^i(2) = 4.00 + 0.00 + 5.20 = 9.20 \\
\theta_3^o(3) &= \min_{2 \le SI \le 3} \{c_3(3, SI)\} = 7.41 \\
&\quad c_3(3,2) = 2\sqrt{2}\sqrt{2+1-3} + \theta_1^o(2) + \theta_2^i(3) = 0.00 + 1.41 + 6.00 = 7.41 \\
&\quad c_3(3,3) = 2\sqrt{2}\sqrt{3+1-3} + \theta_1^o(3) + \theta_2^i(3) = 2.83 + 0.00 + 6.00 = 8.83 \\
\theta_3^o(4) &= \min_{3 \le SI \le 3} \{c_3(4, SI)\} = 6.71 \\
&\quad c_3(4,3) = 2\sqrt{2}\sqrt{3+1-4} + \theta_1^o(3) + \theta_2^i(4) = 0.00 + 0.00 + 6.71 = 6.71
\end{aligned}$$

Finally, we compute $\theta_4^i(SI)$ for $SI = 0, \ldots, M_4 - T_4 = 4$. Again, 4 is a demand stage, so the minimum ranges only over $S = 1$. However, we need to take greater care with the minimization in (6.50) since $\theta_3^o(x)$ is not monotonic in x.

$$\begin{aligned}
\theta_4^i(0) &= \min_{S=1}\{c_4(S,0)\} = c_4(1,0) = 3\sqrt{0+1-1} + \min_{0 \le x \le 0}\{\theta_3^o(x)\} \\
&= 0.00 + \theta_3^o(0) = 0.00 + 8.28 = 8.28 \\
\theta_4^i(1) &= \min_{S=1}\{c_4(S,1)\} = c_4(1,1) = 3\sqrt{1+1-1} + \min_{0 \le x \le 1}\{\theta_3^o(x)\} \\
&= 3.00 + \theta_3^o(1) = 3.00 + 6.69 = 9.69 \\
\theta_4^i(2) &= \min_{S=1}\{c_4(S,2)\} = c_4(1,2) = 3\sqrt{2+1-1} + \min_{0 \le x \le 2}\{\theta_3^o(x)\} \\
&= 4.24 + \theta_3^o(1) = 4.24 + 6.69 = 10.93 \\
\theta_4^i(3) &= \min_{S=1}\{c_4(S,3)\} = c_4(1,3) = 3\sqrt{3+1-1} + \min_{0 \le x \le 3}\{\theta_3^o(x)\} \\
&= 5.20 + \theta_3^o(1) = 5.20 + 6.69 = 11.89 \\
\theta_4^i(4) &= \min_{S=1}\{c_4(S,4)\} = c_4(1,4) = 3\sqrt{4+1-1} + \min_{0 \le x \le 4}\{\theta_3^o(x)\} \\
&= 6.00 + \theta_3^o(1) = 6.00 + 6.69 = 12.69
\end{aligned}$$

The minimum value is $\theta_4^i(0) = 8.28$, so 8.28 is the optimal cost. The optimal solution has an inbound time of 0 to stage 4, which means $S_3^* = 0$. Since $\theta_3^o(0)$ is minimized when $SI = 0$, the inbound time to stage 3 is 0, hence $S_1^* = 0$. The optimal solution is therefore $\mathbf{S}^* = (0, 0, 0, 1)$. The safety stock at each stage is

$$\begin{aligned} SS_1 &= \sqrt{2}\sqrt{1+2-0} = 2.45 \\ SS_2 &= \sqrt{0+1-0} = 1.00 \\ SS_3 &= \sqrt{2}\sqrt{0+1-0} = 1.41 \\ SS_4 &= \sqrt{0+1-1} = 0.00 \end{aligned}$$

Note that the safety stock is pushed upstream as far as possible: Stage 2 needs to hold *some* safety stock since its processing time is 1 and its CST is 0. Since the holding cost at stages 2 and 4 is high, it is important for stage 3 to quote a CST of 0, so it, too, must hold safety stock. But the bulk of the safety stock is held at stage 1 since the holding cost is smallest there. Stage 1, then, absorbs most of the demand uncertainty by serving as the supply chain's main buffer. □

6.4 CLOSING THOUGHTS

As we discussed at the start of this chapter, one can view the stochastic- and guaranteed-service models as two approaches for optimizing base-stock levels in the same system—two algorithms for the same problem. On the other hand, the two models treat backorders in very different ways: The stochastic-service model expects the system to provide instant service to the end-customer and imposes a stockout cost at a rate of p per unit, starting as soon as the customer arrives and finds the product out of stock and continuing as long as is required to clear the backorder. The guaranteed-service model, on the other hand, allows backorders to occur, for free, for up to S time periods and then disallows them entirely after that. (See Figure 6.2.)

This difference causes the guaranteed-service approach to generate solutions in which only a few stages hold inventory, absorbing the uncertainty on behalf of the entire supply chain. The stages that hold inventory act as *push* (or *make-to-stock*) systems, while those that hold no inventory operate as *pull* (or *make-to-order*) systems. In a push system, inventory is produced based on a demand forecast, in anticipation of actual demands. In a pull system, in contrast, production does not begin until a demand triggers, or "pulls," the production process; a pull system holds little or no inventory.

To see this play out in the SSSPP model, let's return to the Kodak supply chain pictured in Figure 6.6. In Figure 6.14, we have indicated hypothetical processing times, T_i, below each stage and holding costs, h_i, (in cents) above. Assume the demand standard deviation is $\sigma = 10$ and $\alpha = 0.95$. Each time period lasts 1 week. The CST for the final stage (Build/test/pack) is a constant, $s = 2$.

The optimal CSTs for this system, obtained using Algorithm 6.2, are noted on the arcs in Figure 6.15. The buckets above each node depict the inventory level at that node. The final stage (Build/test/pack) holds no inventory: It receives inbound CSTs of $SI = 0$ from its suppliers, has a processing time of $T = 2$, and gives its customer a CST of $S = 2$, for an NLT of 0. The stages immediately upstream from Build/test/pack hold inventory so that they can provide inventory on demand to Build/test/pack. The stages farther upstream also

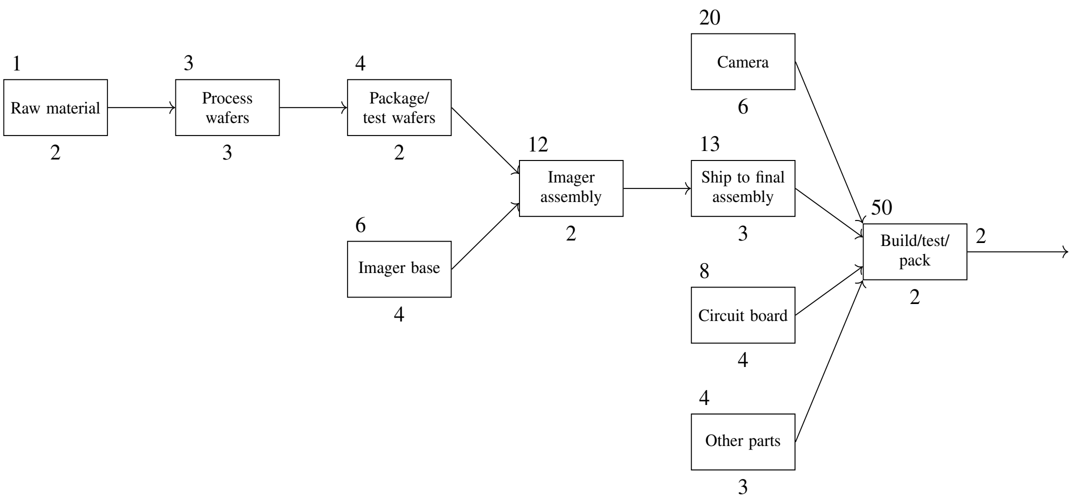


Figure 6.14 Digital camera supply chain network, with holding costs and processing times.

hold no inventory: Each quotes an outbound lead time equal to the sum of its inbound lead time and its processing time. The exception is the Raw material stage, which again holds inventory so it can provide quick service.

The Camera, Ship to final assembly, Circuit board, and Other parts stages serve as the *push–pull boundary* in this system. Upstream from this boundary, the system operates as a push system, producing inventory to hold in anticipation of future demands. Downstream from the boundary is a pull system, in which no production is undertaken until an actual demand has been realized.

The push–pull boundary will change as the system parameters change. For example, suppose the CST promised to the end customer is 8 instead of 2. This gives the downstream portion of the supply chain more time to react to demands, allowing inventory to be stored further upstream, where it is cheaper. (See Figure 6.16.) This also moves the push–pull boundary upstream.

Figure 6.17 plots the expected holding cost as a function of the end-customer CST, s_1. The sharper jumps in the curve (for example, at $s_1 = 2$ and 8) correspond to changes in safety stock locations, while the smoother movements along the curve correspond to changes in safety stock levels. One can view this as a *trade-off curve* that allows the decision maker to navigate the two competing objectives of service and cost. Note that when $s_1 \geq 14$, the entire supply chain can operate as a pull system, holding no inventory and incurring no costs.

The guaranteed-service model is particularly adept at deciding whether stages should operate in push or pull mode since it tends to generate solutions in which only a subset of the stages hold inventory. For instance, suppose we reduce the CST of the Imager assembly stage in Figure 6.15 so that Ship to final assembly holds less inventory and Imager assembly begins to hold some. This means increasing the NLT at Imager assembly and decreasing it at Ship to final assembly. Since the safety stock is a concave function of the NLT, increasing the NLT at Imager assembly has a larger impact on the objective function than does decreasing the NLT at Ship to Final assembly from 10. This makes it unlikely to be a cost-effective change, unless the holding cost is much cheaper at Imager assembly.

In contrast, the objective of the stochastic-service model is a convex function of the base-stock level of each stage (see Section 6.2.2), encouraging the inventory to be more evenly distributed throughout the system.

Although the stochastic- and guaranteed-service models describe the system in different ways and produce different sets of base-stock levels, it is important to note that these are modeling differences rather than operational ones. That is, once we set the base-stock levels, the system operates the same, whether those base-stock levels were set using the stochastic- or guaranteed-service approach. In the guaranteed-service model, there is no need to impose an operational rule requiring orders to be shipped within S periods; the CSTs will automatically be satisfied as a result of the base-stock levels and the demand bound. (See Problem 6.12.) And in the stochastic-service model, there is no need to require demands to be satisfied from stock whenever possible; that, too, will happen as a result of the base-stock levels.

Which model we choose depends on how accurately each one models the particular features of the real-world system and how tractable each one is. In our experience, stochastic-service models tend to be a more natural way to describe most real-world supply chains (since managers are more accustomed to thinking in terms of inventory levels than in terms of CSTs). On the other hand, guaranteed-service models are typically much more

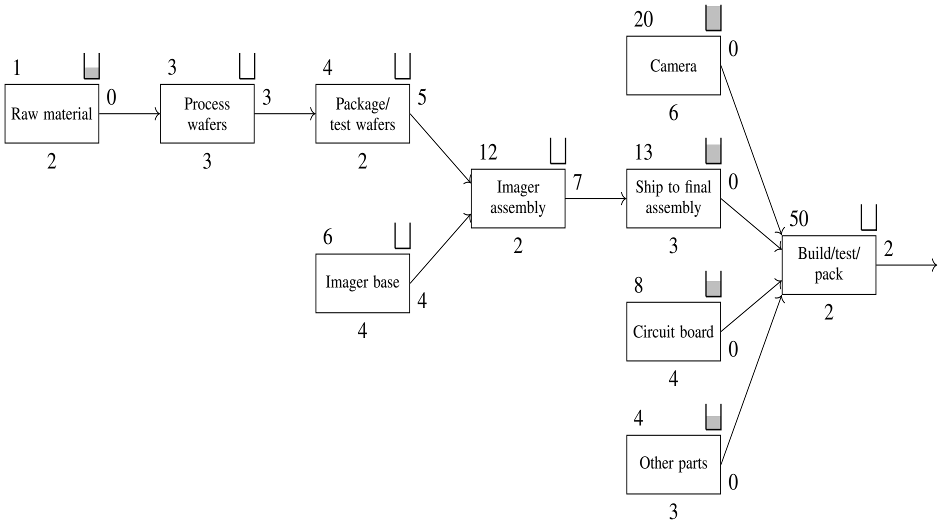


Figure 6.15 Optimal CSTs and inventories for digital camera supply chain (CST to end customer = 2).

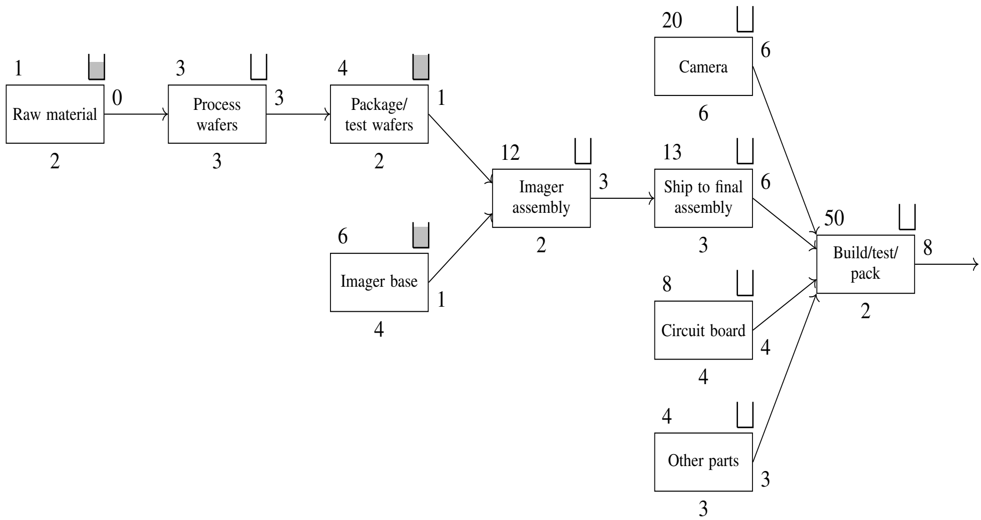


Figure 6.16 Optimal CSTs and inventories for digital camera supply chain (CST to end customer = 8).

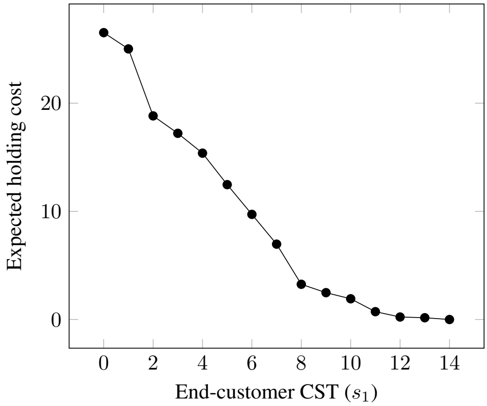


Figure 6.17 SSSPP trade-off curve: expected cost vs. end-customer CST.

tractable and have therefore been implemented in more commercial software packages for multiechelon inventory optimization than stochastic-service models have.

CASE STUDY 6.1 Multiechelon Inventory Optimization at Procter & Gamble

Procter & Gamble (P&G) is one of the world's largest consumer products companies, with annual sales of nearly $80 billion. Their 200 brands include many household names such as Tide laundry detergent, Crest toothpaste, and Gillette razors. P&G is often ranked as one of the best-managed supply chains in the world; for example, in 2016 Gartner named P&G as one of only two companies (along with Apple) in its "Masters" category, for companies whose supply chains are in its top-5 rankings for multiple years (Gartner, Inc. 2016). This is all the more impressive given that the company operates roughly 500 supply chains, consisting of several hundred locations owned both by P&G and by third-party partners.

One of the primary tools that P&G uses to ensure supply chain efficiency is inventory optimization, including both single-stage models, such as those in Chapters 4 and 5, and multiechelon models, especially the SSSPP discussed in Section 6.3. Farasyn et al. (2011) discuss the implementation of both types of models at P&G; we discuss the latter here. (For more information on P&G's single-stage models, see also Farasyn et al. (2008).)

The SSSPP model implemented at P&G has several additional factors that make it more complicated than the model discussed in this chapter. Most significantly, the presence of reorder intervals (see Section 4.3.4.1) and batch production processes destroys the concavity of the safety stock level as a function of the inbound and outbound CSTs (e.g., in (6.46)), and therefore of the objective function. In addition, the Beauty and Grooming supply chains contain (undirected) cycles, so the algorithm for tree networks in Section 6.3.5 does not apply. Instead, P&G used the algorithm of Humair and Willems (2011), which is based on the DP for trees but can solve general systems with nonconcave objective functions to optimality.

A typical Beauty and Grooming SSSPP network (modeling one product family) has 4,000–5,000 stages (representing both locations and processing activities) and 6,000–

10,000 arcs. The 500 or so demand stages in such a network represent multiple finished goods SKUs within the same product family—for example, multiple flavors, sizes, and packaging types for a toothpaste brand. This proliferation of SKUs is the result of a significant push at P&G for postponement (see Section 7.3). A product family discussed by Farasyn et al. (2011) uses a service level (α) of 99.5%, has inbound CSTs (SI) from suppliers of 7 days to 8 weeks, and has production times (T) of 1–2 days and transportation lead times (also modeled using T) of 1–7 days. The demand mean and standard deviation were estimated using the previous 13 weeks of historical data and demand forecasts for the coming 13 weeks.

The multiechelon inventory optimization process resulted in changes to both the locations and quantities of safety stocks in P&G's supply chains. Safety stock levels for raw materials and finished goods decreased, while those for intermediate stages increased. However, the increased cost of intermediate inventory was more than offset by cost reductions for the other inventory types, for a net savings of 17% for the supply chain discussed by Farasyn et al. (2011), and of 7% for the entire North America cosmetics supply chain. This savings is on top of significant savings that had already been achieved through single-stage inventory optimization. And, since it is built into the SSSPP model, the service level to the end customer remained at its target level of 99.5%.

PROBLEMS

6.1 (Exact Algorithm for Serial Systems) Using the exact algorithm for serial systems with stochastic service in Section 6.2.2, find optimal base-stock levels for the following instance: $N = 2$, $p = 15$, $L_1 = L_2 = 1$, $h_1 = h_2 = 1$, and the demand per unit time is distributed $N(100, 15^2)$. Report both echelon and local base-stock levels (S_j^* and $(S')_j^*$).

6.2 (Shang–Song Heuristic) Using the Shang–Song heuristic discussed in Section 6.2.3, find near-optimal base-stock levels for the following instance: $N = 5, p = 24, L_1 = \cdots = L_5 = 0.5$, $h_1 = h_2 = 2$, and $h_3 = h_4 = h_5 = 1$.

- **a)** Assume the demand per unit time is normally distributed with a mean of 64 and a standard deviation of 8.
- **b)** Assume the demand per unit time has a Poisson distribution with $\lambda = 64$.

Report both echelon and local base-stock levels ($\tilde{S}_j$ and $\tilde{S}'_j$).

6.3 (Comparison of Exact and Heuristic Approaches) Find optimal and near-optimal base-stock levels for the following serial system using both the exact approach from Section 6.2.2 and the Shang–Song heuristic from Section 6.2.3: $N = 4$, $p = 80$, $L_1 = \cdots = L_4 = 1$, $h_j = 5 - j$ for all j, and the demand per unit time is distributed $N(20, 4^2)$. Report the echelon base-stock levels and the expected cost of each solution.

6.4 (Proof of Proposition 6.1) Prove Proposition 6.1.

6.5 (Equivalence of Local- and Echelon-Based Total Costs) Prove that $g'(\mathbf{S}')$ in (6.8) equals $g(\mathbf{S})$ in (6.9).

6.6 (Proof of "All-or-Nothing" Theorem) Prove Theorem 6.5.

Note: You may use the fact that there exists an optimal solution in which, for all i, either $S_i = S_{i+1} + T_i$ or $S_i = \max\{0, S_{i-1} - T_{i-1}\}$.

6.7 (Safety Stock for Ceramic Plates) A manufacturer of ceramic plates and other tableware divides the manufacturing process into three major steps: forming, firing, and glazing. In the first step, the plates are formed out of clay; in the second, the plates are heated in a kiln, and in the third, the plates are painted. Forming and firing each take 1 day, while glazing takes 2 days. Clay is procured from an external vendor, which delivers orders exactly 1 day after they are placed. The daily demand for plates, as measured in cases, is distributed $N(45, 10^2)$. The company promises its customers that finished (i.e., glazed) plates will always be on-hand provided that the demand on a given day is no more than 4 standard deviations above its mean. (That is, $s_1 = 0$ and $z_\alpha = 4$.) Inventory may be held at any stage of the process. The holding cost of one case of plates (or its precursor products) is \$2 per day for plates that have been formed but not fired, \$3 for plates that have been fired but not glazed, and \$4 for glazed plates. Find the optimal CST, base-stock level, and safety-stock level at each stage, as well as the optimal expected cost per day.

6.8 (Implementing Serial SSSPP DP) The file `serial10.xlsx` contains the holding costs and processing times for a 10-stage serial system. The demand per period is distributed $N(89.0, 15.8^2)$. Use $\alpha = 0.98$ in the demand bound. There is an inbound service time of 7 periods at stage 10, and stage 1 has a CST of 3 to the customer. Implement the DP algorithm from Section 6.3.4 and use it to find the optimal CST, base-stock level, and safety-stock level at each stage, as well as the optimal expected cost per period.

6.9 (Safety Stock for Baseball Hats) Figure 6.18 depicts the supply chain for a firm that manufactures baseball hats for college baseball fans. There are two end products. Product 1 is a Lehigh University hat, for which the firm sees a daily demand that is normally distributed with a mean of 22.0 cases and a standard deviation of 4.1 cases. Product 2 is a Lafayette College hat, whose demand is also normally distributed, with a mean of 15.3 cases and a standard deviation of 6.2 cases.

Stage 3 represents assembling the hats from two subassemblies: the cap (the part that sits on your head) and the visor (the part that sticks out in front). This generic product is then differentiated at stages 1 and 2 by dyeing the fabric and embroidering the team logos. Stage 4 represents the visor subassembly, while stage 5 represents sewing the cap subassembly out of fabric; the fabric is represented by stage 6.

Figure 6.18 indicates the processing time below each stage and, above it, the value of one case's worth of the product. The firm is committed to providing a CST of 3 days to its customers (such as college bookstores). It has also set CSTs for the upstream stages, which are indicated on the links in the figure, but you suspect that these are not the optimal CSTs.

a) Calculate the base-stock level and safety-stock level required at each stage for the solution in the figure, as well as the total expected holding cost. Assume that demands are truncated 4 standard deviations above their means; i.e., $z_\alpha = 4$ in (6.33). Also assume that holding costs are calculated as 20% of the product value, per year. (Make sure to translate into days.)

b) Develop a solution to the SSSPP that still gives CSTs of 3 days to the end customers but is cheaper than the solution depicted in Figure 6.18. Your solution does not need to be optimal, only better than the one in Figure 6.18. For each

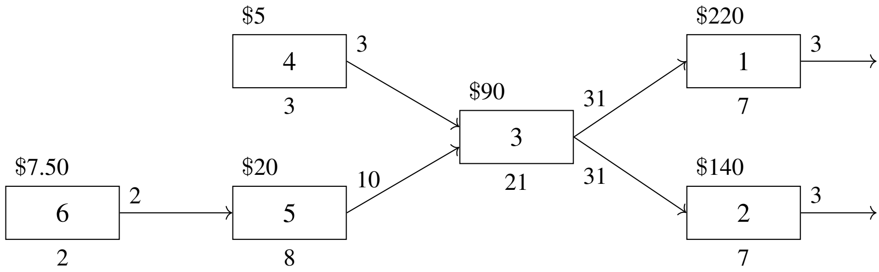


Figure 6.18 Baseball-hat supply chain for Problem 6.9.

stage, report the CST, base-stock level, and safety-stock level, as well as the total expected holding cost per period for the whole system.

6.10 (Implementing Tree SSSPP DP) Implement Algorithm 6.2 and use it to find the optimal solution for the instance introduced in Problem 6.9. Report the optimal CST, base-stock level, and safety-stock level at each stage, as well as the optimal expected cost per period.

6.11 (Two-Stage SSSPP) Consider a two-stage serial supply chain with guaranteed service as defined in Section 6.3.4. Assume that $0 < h_2 < h_1$. The inbound CST to stage 2, SI_2, is a constant, as is the outbound CST from stage 1, s_1. Therefore, the only decision variable is S_2. For simplicity, assume that $z_\alpha = \sigma = 1$. Then the objective function is given by

$$g(S_2) = h_2\sqrt{SI_2 + T_2 - S_2} + h_1\sqrt{S_2 + T_1 - s_1}.$$

a) Prove that, in the optimal solution to the SSSPP for the two-stage supply chain defined above:

i. Stage 1 holds safety stock if and only if $s_1 < T_1$.

ii. If stage 1 holds safety stock, then stage 2 also holds safety stock if and only if

$$h_2\sqrt{SI_2 + T_2} + h_1\sqrt{T_1 - s_1} < h_1\sqrt{SI_2 + T_2 + T_1 - s_1}.$$

b) Now consider an N-stage serial supply chain, with SI_N and s_1 constants, as usual. Assume that $0 < h_N < \cdots < h_1$. Prove that if

$$\sum_{i=1}^{k} T_i < s_1,$$

then stages $1, 2, \ldots, k$ hold no safety stock.

6.12 (CSTs are Satisfied) Simulate a single-stage system under the guaranteed-service model in a programming language or spreadsheet program of your choice. Assume $SI = 4$, $T = 2$, and $S = 3$. Assume the demand per period is distributed as $N(50, 10^2)$ and use $z_\alpha = 1.5$ to truncate the demand. Use the appropriate base-stock level y for these settings and assume the system begins period 1 with y units on hand. Assume that demands are satisfied first-come, first-served. Simulate the system for at least 1000 periods and verify that, as claimed in Section 6.4, the CST is always satisfied, even though your simulation does not contain explicit logic to ensure it.

Hint: Include columns or variables that keep track of the number of unsatisfied demands that were placed 0, 1, 2, 3, and 4 or more periods ago.

6.13 (Limits of $\hat{g}_j$ Function) In the exact algorithm for serial systems described in Section 6.2.2, prove that, for all $j = 1, \ldots, N$:

$$\lim_{x \to -\infty} \hat{g}_j(x) = \sum_{i=1}^{j} h_i \left(x - \sum_{k=i}^{j-1} \mathbb{E}[D_k] \right) - (p + h'_1) \left(x - \sum_{k=1}^{j-1} \mathbb{E}[D_k] \right) \tag{6.53}$$

$$\lim_{x \to +\infty} \hat{g}_j(x) = \begin{cases} h_j x + g_{j-1}(S^*_{j-1}), & \text{if } j > 1 \\ h_j x, & \text{if } j = 1 \end{cases} \tag{6.54}$$

where $\sum_{k=a}^{b}[\text{anything}] \equiv 0$ if $a > b$. What types of functions are these (quadratic, linear, concave, etc.)?

6.14 (Proof of (6.13)**)** In this problem you will prove (6.13). Throughout this problem, you may use any of the results up until (6.12), but nothing that comes later.

a) Prove that

$$IL'_j(t) = IL_j(t) - IP_{j-1}(t).$$

b) Use part (a) to prove that

$$ITP_j(t) = \min\{S_j, IL_{j+1}(t)\}.$$

6.15 (Approximate Two-Stage SSM Model) Consider a 2-stage serial system following an echelon base-stock policy under the SSM model. The costs, demand rate, and lead times are as given in Section 6.2.1. Assume the demand per unit time is distributed as $N(\mu, \sigma^2)$, so the lead-time demand for stage j has a mean of $\mu_j \equiv \mu L_j$ and a standard deviation of $\sigma_j \equiv \sigma\sqrt{L_j}$.

In this problem, you will develop an approximate method for computing the cost of a given echelon base-stock policy. This method is much easier to implement than the Clark–Scarf recursion in Theorem 6.3.

From (6.9), the expected cost for a given echelon base-stock policy $\mathbf{S} = (S_1, S_2)$ can be written

$$g(\mathbf{S}) = E[h_1 IL_1 + h_2 IL_2 + (p + h'_1) IL_1^-].$$

This expression has three random variables; you'll use exact expressions for the expectations of the first two and develop an approximation for the third.

From (6.30),

$$S^*_1 = \mu_1 + \sigma_1 \Phi^{-1} \left(\frac{p + h_2}{p + h'_1} \right). \tag{6.55}$$

At stage 2, $ITP_2 = IP_2 = S_2$ since stage 2's supplier never has stockouts. Therefore, from (6.16),

$$E[IL_2] = S_2 - \mu_2 \tag{6.56}$$

$$E[IL_1] = E[ITP_1] - \mu_1. \tag{6.57}$$

a) Prove that

$$E[ITP_1] = S_1 - \sigma_2 \mathscr{L} \left(\frac{S_2 - S_1 - \mu_2}{\sigma_2} \right), \tag{6.58}$$

where $\mathscr{L}(\cdot)$ is the standard normal loss function. You may assume $S_2 \geq S_1$.

b) From (6.16), $IL_1 = ITP_1 - D_1$. To calculate $E[IL_1^-]$ exactly, therefore, we need to use the distribution of ITP_1. Unfortunately, this distribution is fairly complicated. The approximation we are proposing instead is to replace the stochastic ITP_1 with its mean, $E[ITP_1]$. (We are not suggesting this is a very good approximation. In general, replacing a random variable with its mean can lead to significant inaccuracy. But it makes the problem more tractable, so we will try it.)

Prove that, under this approximation,

$$E[IL_1^-] = \sigma_1 \mathscr{L}\left(\frac{E[ITP_1] - \mu_1}{\sigma_1}\right). \tag{6.59}$$

c) Do you think the approximation in part (b) will underestimate or overestimate $E[IL_1^-]$? Explain your answer in one or two sentences.

6.16 (Implementing Approximate Two-Stage SSM Model) Implement the approximation from Problem 6.15 in MATLAB to compute the expected cost using the *optimal* S_1 and a *given* value for S_2.

(*Hint*: To double-check that your calculations are correct, we'll tell you the following: If $h_1 = h_2 = 1$, $p = 10$, $L_1 = L_2 = 2$, $\mu = 10$, $\sigma = 3$, $S_1 = 10$, $S_2 = 25$, then $g(\mathbf{S}) = 172.7378$.)

a) Compute the optimal $\mathbf{S}$ for a system with $h_1 = 5$, $h_2 = 2$, $p = 24$, $L_1 = 8$, $L_2 = 3$, $\mu = 20$, and $\sigma = 4$. Use (6.55) to find S_1^*, then find S_2^* in MATLAB using a method of your choosing: trial and error; MATLAB's `fminunc` function; etc. Report S_1^*, S_2^*, and g^*. Include a printout of all MATLAB code, including a transcript of the session in which you found S_2^*.

b) Compute values for the following quantities assuming $\mathbf{S}$ is set to the optimal values from part (a). (*Hint*: You should not have to evaluate any more integrals.)

- The expected on-hand inventory at stage 1, $E[IL_1^+]$.
- The expected backorders at stage 1, $E[IL_1^-]$.
- The expected inventory level at stage 1, $E[IL_1]$.
- The expected local on-hand inventory at stage 2, $E[(IL_2')^+]$.
- The expected local backorders at stage 2, $E[(IL_2')^-]$.
- The expected echelon inventory level at stage 2, $E[IL_2]$.
- The expected number of units in transit from stage 2 to stage 1, $E[IT_1]$.
- The expected holding, stockout, and total costs per period.

CHAPTER 7

POOLING AND FLEXIBILITY

7.1 INTRODUCTION

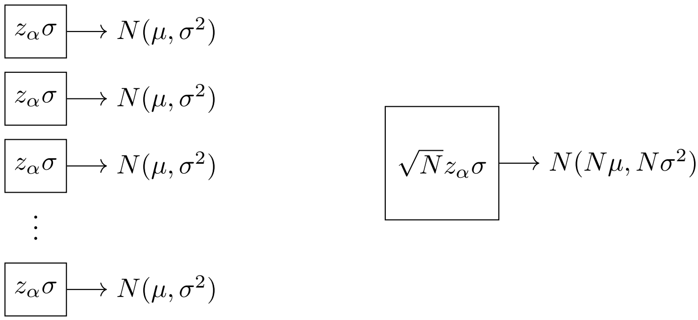
The stochastic inventory models in Chapters 4–6 assume that inventory is the only tool for mitigating uncertainty. In contrast, this chapter examines uncertainty mitigation using other means. In all of the strategies covered here, the idea is to "pool" multiple demand streams in some way, and to share some resource—inventory or capacity—among them. Because not all of the demand streams will need all of the resources at all times, there is no need to dedicate whole resources to each stream. By pooling them, we can reduce the amount of safety stock required to meet a given service level (or increase the service level attained by a given level of safety stock).

Section 7.2 deals with risk pooling, in which we physically combine the inventories used to satisfy multiple demand streams, by storing them together in the same warehouse. Section 7.3 discusses a strategy called postponement, in which we differentiate products later in their manufacturing process. This allows a reduction in inventory since multiple demand streams (from different end products) are sharing inventory of the the undifferentiated product. The cost savings from postponement is due to the risk pooling effect.

Another way that inventory can be pooled is by allowing transshipments—"lateral" transfers of inventory from one retailer to another when one has extra inventory and the other has a shortage. In Section 7.4, we discuss a model for deciding how much inventory to hold at a given retailer, anticipating that transshipments either to or from that retailer

Fundamentals of Supply Chain Theory, Second Edition. Lawrence V. Snyder and Zuo-Jun Max Shen.

Companion website: www.wiley.com/go/Snyder/SupplyChainTheory



(a) Decentralized system. (b) Centralized system.

Figure 7.1 The risk-pooling effect with identical retailers.

may occur later in the period. The benefit from transshipments is similar to that from risk pooling, although there is no physical pooling of inventory.

Similarly, when multiple products are sold, and each product is manufactured by a dedicated plant, it is sometimes beneficial for one plant to make multiple products so that when one product has very high demand (exceeding the capacity of a plant that makes the product), other plants can help produce more units of the product to meet the demand. In this case, we have "lateral" transfers of production capacity, in a strategy known as process flexibility. We can think of process flexibility as a type of pooling that occurs when the product is manufactured, rather than when it is stored. We discuss process flexibility in Section 7.5.

7.2 THE RISK-POOLING EFFECT

7.2.1 Overview

Consider a network consisting of N distribution centers (DCs) or other facilities, each of which faces random demand for a single product. The DCs each hold inventory of this product. In fact, they act like N independent newsvendors, each facing $N(\mu, \sigma^2)$ demand per period. If the DCs each wish to meet a type-1 service level of α (that is, they wish to stock out in no more than $100(1-\alpha)\%$ of the periods on average), they must each hold an amount of safety stock equal to $z_\alpha\sigma$ (from (4.24)). The total safety stock in this system is therefore $Nz_\alpha\sigma$. (See Figure 7.1(a).)

Now suppose that all N DCs are merged into a single DC. What are the inventory implications of this consolidation? (We're ignoring the possible increase in transportation cost, lead time, and hassle the consolidation may cause.) The new DC's demand process is equal to the sum of all of the original DCs' demands. This process has a mean demand of $N\mu$ and a standard deviation of $\sqrt{N}\sigma$. Therefore, to meet the same service level (α), the new DC needs to hold $\sqrt{N}z_\alpha\sigma$ of safety stock (see Figure 7.1(b)), which is less than the safety stock required when N DCs each hold inventory.

This phenomenon is known as the *risk-pooling effect* (Eppen 1979). The basic idea is that by pooling demand streams, we can reduce the amount of safety stock required to meet a given service level, and hence, we can reduce the holding cost.

We next discuss the risk-pooling effect in greater generality. Our analysis is adapted from that of Eppen (1979).

7.2.2 Problem Statement

We'll assume that each DC follows a base-stock inventory policy under periodic review, with S_i the base-stock level for DC i. The lead time is $L = 0$ at every DC. Excess inventory may be stored from period to period (with a holding cost of h per unit per period), and excess demand is backordered (with a penalty cost of p per unit per period). We assume $p > h$. Note that h and p are the same at every DC.

The demand per period seen by DC i is represented by the random variable D_i, with $D_i \sim N(\mu_i, \sigma_i^2)$. Let f_i and F_i be the pdf and cdf, respectively, of D_i. Demands may be correlated among DCs. The covariance of D_i and D_j is given by σ_{ij} and the correlation coefficient by ρ_{ij}; then $\sigma_{ij} = \sigma_i \sigma_j \rho_{ij}$. (Corbett and Rajaram (2006) extend these results to general probability distributions and dependence structures.)

For each DC, the sequence of events in each period is the same as in Section 4.3.

7.2.3 Decentralized System

We will refer to the N-DC system as the *decentralized system* since each DC operates independently of the others. S_i is the base-stock level at DC i; this is a decision variable. The expected cost per period at DC i can be expressed as a function of S_i as follows:

$$g_i(S_i) = h \int_0^{S_i} (S_i - d) f_i(d) dd + p \int_{S_i}^{\infty} (d - S_i) f_i(d) dd.$$

This formula is identical to the formula for the newsvendor cost (4.3) except for the subscripts i. Therefore, from Theorems 4.1 and 4.2, the optimal solution is

$$S_i^* = F_i^{-1}\left(\frac{p}{h+p}\right) = \mu_i + z_\alpha \sigma_i,$$

where $\alpha = p/(p+h)$ and z_α is the αth fractile of the standard normal distribution, and the optimal cost at DC i is

$$g_i(S_i^*) = (p+h)\phi(z_\alpha)\sigma_i.$$

(Recall that $\phi(\cdot)$ is the pdf of the standard normal distribution.) Defining $\eta = (p+h)\phi(z_\alpha)$ for convenience, the optimal total expected cost (at all DCs) in the decentralized system, denoted g_D^*, is

$$g_D^* = \sum_{i=1}^N g_i(S_i^*) = \eta \sum_{i=1}^N \sigma_i. \tag{7.1}$$

7.2.4 Centralized System

Now imagine that the DCs are consolidated into a single DC, denoted with index 0, that serves all of the demand. We will refer to this as the *centralized system*. Let D_0 be the total demand seen by this super-DC. Its mean and standard deviation are

$$\mu_0 = \sum_{i=1}^N \mu_i$$

$$\sigma_0 = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N}\sigma_{ij}}.$$

(Note that by definition, $\sigma_{ii} = \sigma_i^2$.) Similar logic as above shows that the optimal base-stock level for the centralized system is

$$S_0^* = \mu_0 + z_\alpha\sigma_0$$

with optimal expected cost

$$g_C^* = \eta\sigma_0 = \eta\sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N}\sigma_{ij}}. \tag{7.2}$$

7.2.5 Comparison

Now let's compare the centralized and decentralized systems. The next theorem says that the centralized system is no more expensive than the decentralized system. This is the risk-pooling effect.

Theorem 7.1 *For the decentralized, N-DC system and the centralized, single-DC system formed by merging the DCs, $g_C^* \le g_D^*$.*

Proof.

$$\begin{aligned}
g_C^* &= \eta\sqrt{\sum_{i=1}^{N}\sigma_i^2 + 2\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\sigma_i\sigma_j\rho_{ij}} \\
&\le \eta\sqrt{\sum_{i=1}^{N}\sigma_i^2 + 2\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\sigma_i\sigma_j} \qquad \text{(since } \rho_{ij} \le 1\text{)} \\
&= \eta\sqrt{\left(\sum_{i=1}^{N}\sigma_i\right)^2} \\
&= g_D^*.
\end{aligned}$$

■

One interpretation of the risk-pooling effect is that pooling inventory allows the firm to take advantage of random fluctuations in demand. If one DC sees unusually high demand in a given time period, it's possible that another DC sees unusually low demand. In the centralized system, the excess inventory at the low-demand DC can be used to make up the shortfall at the high-demand DC. In the decentralized system, there is no opportunity for this supply–demand matching.

A more mathematical explanation is that risk pooling occurs because the centralized system takes advantage of the concave nature of safety stock requirements. The amount of safety stock required is proportional to the standard deviation of demand. The standard deviation of demand at the centralized site is smaller than the sum of the standard deviations

Table 7.1 Demand mean and standard deviation at DCs in Example 7.1.

i	μ_i	σ_i
1	32,500	6200
2	18,200	1100
3	21,000	5900
4	11,400	1400
5	29,300	4200

of the individual sites in the decentralized system since variances, not standard deviations, are additive.

Somewhat surprisingly, the *variances* of the costs of the centralized and decentralized systems are equal at optimality; that is,

$$\text{Var}\left[\hat{g}_D(S_1^*, \ldots, S_N^*)\right] = \text{Var}\left[\hat{g}_C(S_0^*)\right],$$

where

$$\hat{g}_D(S_1, \ldots, S_N) = \sum_{i=1}^{N} \left[h(S_i - D_i)^+ + p(D_i - S_i)^+\right]$$
$$\hat{g}_C(S_0) = h(S_0 - D_0)^+ + p(D_0 - S_0)^+$$

are the costs in the decentralized and centralized systems, respectively, for given (random) values of the demands.

☐ **EXAMPLE 7.1**

Gauss & Poisson manufactures household cleaners, beauty products, facial tissues, and other consumer packaged goods (CPG). G&P currently operates five DCs. The mean and standard deviation of the demand served by each DC per month, expressed in thousands of pallets, is listed in Table 7.1. The demands at the five DCs are normally distributed and are independent of one another. Each pallet of inventory incurs a holding cost of \$1.30 per month, and each pallet of backordered demand incurs a stockout cost of \$17.50 per month. What is the optimal total expected cost at the DCs? Suppose G&P decides to merge the five DCs into a single DC. What is the new optimal expected cost?

We have $\alpha = p/(p+h) = 0.9309$ and $z_\alpha = 1.4822$. Therefore, $\eta = (17.5 + 1.3)\phi(1.4822) = 2.5006$. Under the decentralized system (five DCs), the total expected cost, from (7.1), is

$$g_D^* = 2.5004 \cdot (6200 + 1100 + 5900 + 1400 + 4200) = 47{,}010.78.$$

If the five DCs are merged, the resulting standard deviation of demand is

$$\sigma_0 = \sqrt{6200^2 + 1100^2 + 5900^2 + 1400^2 + 4200^2} = 9698.4535,$$

so the new expected cost, from (7.2), is

$$g_C^* = 2.5004 \cdot 9698.4534 = 24{,}251.69,$$

significantly smaller than g_D^*. ☐

7.2.6 Magnitude of Risk-Pooling Effect

Let's try to get a handle on the magnitude of the risk-pooling effect. Let

$$v = 2\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\sigma_i\sigma_j\rho_{ij}.$$

Note that

$$g_C^* = \eta\sqrt{\sum_{i=1}^{N}\sigma_i^2 + v}.$$

Uncorrelated Demands: First assume that the demands are uncorrelated, i.e., $\rho_{ij} = 0$ for all $i,\ j$, so $v = 0$. Then

$$g_C^* = \eta\sqrt{\sum_{i=1}^{N}\sigma_i^2 + v} = \eta\sqrt{\sum_{i=1}^{N}\sigma_i^2} \le \eta\sqrt{\left(\sum_{i=1}^{N}\sigma_i\right)^2} = g_D^*.$$

The magnitude of the difference between g_C^* and g_D^* depends on the magnitude between $\sqrt{\sum\sigma_i^2}$ and $\sum\sigma_i$.

Positively Correlated Demands: Next suppose that demands are positively correlated. In fact, consider the extreme case in which $\rho_{ij} = 1$ for all $i,\ j$. Then

$$\begin{aligned} g_C^* &= \eta\sqrt{\sum_{i=1}^{N}\sigma_i^2 + v} = \eta\sqrt{\sum_{i=1}^{N}\sigma_i^2 + 2\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\sigma_i\sigma_j} \\ &= \eta\sqrt{\left(\sum_{i=1}^{N}\sigma_i\right)^2} = \eta\sum_{i=1}^{N}\sigma_i = g_D^*, \end{aligned}$$

so there is no risk-pooling effect at all (in the extreme case of perfect correlation).

Negatively Correlated Demands: Finally, assume that demands are negatively correlated. It's difficult to identify the extreme case since ρ_{ij} can't equal -1 for all $i,\ j$. (Why?) But we can say that $v \ge -\sum_{i=1}^{N}\sigma_i^2$ since

$$\sum_{i=1}^{N}\sigma_i^2 + v = \sigma_C^2 \ge 0.$$

So let's assume as an extreme scenario that $v = -\sum_{i=1}^{N}\sigma_i^2$. Then

$$g_C^* = \eta\sqrt{\sum_{i=1}^{N}\sigma_i^2 - \sum_{i=1}^{N}\sigma_i^2} = 0.$$

The centralized cost is 0, while the decentralized cost is not.

So the risk-pooling effect is very pronounced when demands are negatively correlated, smaller when demands are uncorrelated, and smaller still, or even non-existent, when demands are positively correlated. Why? Recall the explanation given in Section 7.2.5: The risk-pooling effect occurs because excess inventory at one DC can be used to meet excess demand at another. If demands are negatively correlated, there is a lot of opportunity to do this since demands will be very disparate at different locations. On the other hand, if demands are positively correlated, they tend to be all high or all low at the same time, so there is little opportunity for supply–demand rebalancing.

☐ **EXAMPLE 7.2**

In Example 7.1, we assumed that G&P's demands are independent. Suppose instead that the demands are positively correlated, with correlation matrix

$$\boldsymbol{\rho} = \begin{bmatrix} 1.0 & 0.3 & 0.9 & 0.7 & 0.7 \\ 0.3 & 1.0 & 0.5 & 0.3 & 0.3 \\ 0.9 & 0.5 & 1.0 & 0.8 & 0.7 \\ 0.7 & 0.3 & 0.8 & 1.0 & 0.7 \\ 0.7 & 0.3 & 0.7 & 0.7 & 1.0 \end{bmatrix},$$

or that some of the demands are negatively correlated, with correlation matrix

$$\boldsymbol{\rho} = \begin{bmatrix} 1.0 & -0.3 & 0.0 & 0.0 & -0.7 \\ -0.3 & 1.0 & 0.0 & 0.0 & 0.5 \\ 0.0 & 0.0 & 1.0 & -0.6 & 0.0 \\ 0.0 & 0.0 & -0.6 & 1.0 & 0.0 \\ -0.7 & 0.5 & 0.0 & 0.0 & 1.0 \end{bmatrix}.$$

How does the magnitude of the risk-pooling effect compare among these three cases?

From Example 7.1, we have $g_D^* = 47{,}010.78$ in all three cases and $g_C^* = 24{,}251.69$ for the independent-demand case.

In the case of positive correlation, we have $v = 1.8487 \times 10^8$, so

$$g_C^* = 2.5006\sqrt{9.4060 \times 10^7 + 1.8487 \times 10^8} = 41{,}762.57.$$

And in the case of negative correlation, $v = -4.5840 \times 10^7$, so

$$g_C^* = 2.5006\sqrt{9.4060 \times 10^7 - 4.5840 \times 10^7} = 17{,}364.14.$$

Thus, as expected, the risk-pooling effect is greatest when the demands are negatively correlated, smallest when they are negatively correlated, and in between when they are independent. ☐

7.2.7 Closing Thoughts

The analysis above only considers holding and stockout costs; it does not consider fixed costs (to build and operate DCs) or transportation costs. Clearly, as DCs are consolidated, the fixed cost will decrease. But the transportation cost will increase, since retailers (or other downstream facilities) will be served from more distant DCs. In many cases, the magnitude of the risk-pooling effect may be far outweighed by the increases or decreases in fixed and transportation cost. Any analysis of a potential consolidation of DCs must include all factors, not just risk pooling. The location model with risk pooling (LMRP), discussed in Section 12.2, attempts to incorporate all of these factors when choosing facility locations.

7.3 POSTPONEMENT

Many firms have product lines containing closely related products. In many cases, multiple end products are made from a single generic product. For example, the clothing retailer Benetton sells many colors of sweater, each of which comes from the same white sweater that's dyed multiple colors (Heskett and Signorelli 1984). Hewlett-Packard sells the same printer in dozens of countries, with a different power supply module, manual, and labels in each (Feitzinger and Lee 1997, Lee and Billington 1993). IBM builds individualized computers by building partially finished products called "vanilla boxes" and customizing them to order (Swaminathan and Tayur 1998).

A key question in the design of the manufacturing process for each of these products is: When should the end products be differentiated? For example, consider a manufacturer of mobile phones that sells phones in many countries. The company programs each phone with a given language at the factory—the phone is "localized" when it is manufactured. The number of phones to be programmed in each language is determined based on a forecast of the demand in each country. The phones are then shipped to regional DCs, approximately one on each continent. The regional DCs store the phones until they are required by retailers, at which point they are shipped to individual countries. If the demand forecasts were wrong, and demand for phones in, say, Thailand was higher than expected while demand in Holland was lower than expected, the company would have to correct this discrepancy by reprogramming some of the Dutch phones into Thai phones, then shipping them from the Europe DC to the Asia DC—a costly and time-consuming proposition.

Now suppose that *generic* phones are shipped to the regional DCs, and languages are programmed at the DCs once the phones are requested by retailers. Since the phones are localized on demand, there is much less risk of having too many phones of one language and too few of another. In addition, the firm holds inventory of generic phones, not localized phones, which means that fewer phones need to be held in safety stock due to the risk-pooling effect, as we will see below.

This strategy is called *postponement* or *delayed differentiation*. The idea is to delay, as much as possible, the point in the manufacturing process at which end products are differentiated from one another. Of course, designing a postponement strategy may be extremely complicated, since it may require the redesign of the product and the manufacturing and distribution processes. In the mobile phone example, the regional DCs would have to be outfitted with language-programming equipment.

To take the Benetton example to an extreme, postponement might mean that sweaters are dyed in the retail stores once they are demanded by a customer. You would request, say, a red sweater, and it would be dyed for you on demand; stores would never be out of stock of the sweater you wanted. This seems silly, since the costs of implementing such a system would probably far outweigh the benefits. But some products are actually sold this way. For example, paint is mixed to order from generic white paint at your hardware store, giving you access to an enormous range of colors that would be prohibitively expensive to keep in stock. (See Lee (1996) for a discussion of the benefits and challenges of postponement.)

The cost savings from postponement is due to the risk-pooling effect: Generic products represent pooled inventory, while end products represent decentralized inventory. Suppose there are N end products. If the products are differentiated at the beginning of the manufacturing process (so that separate inventory is held of each end product), then the total safety stock required is

$$z_\alpha \sum_{i=1}^{N} \sigma_i,$$

which is proportional to the safety stock required in the decentralized system in our discussion of risk pooling. Similarly, if the products are differentiated at the end (so that only a single inventory pile is required), the total safety stock is

$$z_\alpha \sqrt{\sum_{i=1}^{N} \sigma_i^2},$$

which is proportional to the safety stock in the centralized system.

7.4 TRANSSHIPMENTS

7.4.1 Introduction

When multiple retailers stock the same product, it is sometimes advantageous for one retailer to ship items to another if the former has a surplus and the latter has a shortage. Such "lateral" transfers are called *transshipments*. Transshipments are a mechanism for improving service levels since they allow demands to be satisfied in the current period when they might otherwise be lost or backordered until the following period. In that regard, the benefit from transshipments is very similar to that from risk pooling, since transshipments use one retailer's surplus to reduce another retailer's shortfall. In this case, however, there is no physical pooling of inventory, though the strategy is sometimes referred to as "information pooling." Of course, transshipments come at a cost: Transshipments are often more expensive than replenishments from the DC because they are smaller and therefore lack the economies of scale from larger shipments.

In this section, we will discuss a model for setting base-stock levels in a system with two retailers that may transship to one another. This model is adapted from Tagaras (1989). For models with more than two retailers, see Krishnan and Rao (1965), Tagaras (1999), or Herer et al. (2006).

This model will assume that transshipments occur after the demand has been realized but before it must be satisfied. Therefore, these transshipments are *reactive* since they are made in reaction to realized demands. In contrast, one might consider *proactive* transshipments that are made in anticipation of demand shortages. Proactive transshipments are of interest when demands must be met instantaneously, since there is no opportunity for transshipping between demand realization and satisfaction. On the other hand, proactive transshipments are more complex to model, so we will focus only on reactive transshipments. We will develop an analytical expression for the expected cost function, but the expected cost can only be minimized using numerical methods (rather than using differentiation). We will also discuss the improvement in service levels due to transshipments.

7.4.2 Problem Statement

Consider a system with two retailers served by a single DC. The retailers receive *replenishment* shipments from the DC and are permitted to *transship* goods to each other. As previously stated, transshipment occurs after the demand has been realized but before it must be satisfied. This is a periodic-review model with an infinite horizon. There is no fixed cost and no lead time, either for replenishments or transshipments. Each retailer i ($i = 1, 2$) follows a base-stock policy, with base-stock level S_i. The demand at retailer i is a random variable D_i with pdf f_i and cdf F_i. If there are excess demands at a retailer after transshipments have been made, they are backordered. The costs are as follows:

c_i = ordering cost per unit at retailer i, for $i = 1, 2$
h_i = holding cost per unit per period at retailer i, for $i = 1, 2$
p_i = backorder cost per unit per period at retailer i, for $i = 1, 2$
c_{ij} = cost per unit to transship from i to j, for $i = 1, 2,\ i \neq j$

We will assume that

$$c_i - c_j + c_{ij} \geq 0. \tag{7.3}$$

In other words, it is cheaper to ship directly to j than to ship to i and then transship to j. This is sometimes referred to as a *triangle inequality*. We will also make the following assumptions:

(a) $h_i + p_j - c_{ij} - (c_i - c_j) \geq 0$ (i.e., if there is a shortage at j and a surplus at i, it is better to transship than not to, since the cost to transship is c_{ij}, while the cost to do nothing is $h_i + p_j + c_j - c_i$ (since we would incur the holding cost at i, the penalty cost at j, and then next period we'd order one more unit at j and one fewer at i))

(b) $c_{ij} + (c_i - c_j) - (h_i - h_j) \geq 0$ (i.e., don't transship if there is a surplus at both retailers)

(c) $c_{ij} + (c_i - c_j) + (p_i - p_j) \geq 0$ (i.e., don't transship if there is a shortage at both retailers)

These three assumptions imply that *complete pooling* is optimal: Transship if one retailer has a surplus while the other has a shortage, but if both have surpluses or both have shortages don't transship—one retailer's demand is not "more valuable" than the other's.

The sequence of events in each period is as follows:

1. Retailers observe their inventory levels.
2. Each retailer i places a replenishment order of size Q_i to the DC and receives it instantaneously.
3. Demand is observed.
4. Transshipment decisions are made. Transshipments are sent and arrive instantaneously.
5. Demand is satisfied to the extent possible, and excess demands are backordered.
6. Holding and stockout costs are assessed.

We will make use of the following random variables:

Q_i = replenishment order quantity at retailer i, for $i = 1, 2$
Y_{ij} = amount transshipped from i to j, for $i = 1, 2,\ i \neq j$

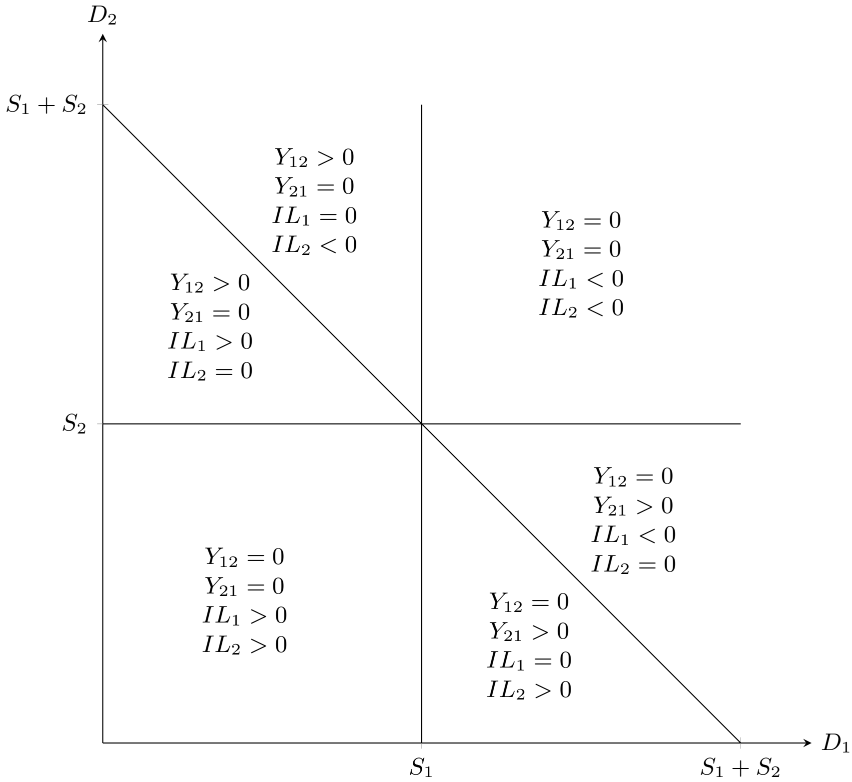


Figure 7.2 Possible realizations of transshipment and ending inventories. Adapted with permission from Tagaras, Effects of pooling on the optimization and service levels of two-location inventory systems, *IIE Transactions*, 21, 1989, 250–257. ©1989, Taylor & Francis, Ltd., `http://www.informaworld.com`.

IL_i = inventory level at retailer i after step 5, for $i = 1, 2$
IL_i^+ = on-hand inventory at retailer i after step 5, for $i = 1, 2$
IL_i^- = backorders at retailer i after step 5, for $i = 1, 2$

Then

$$IL_i = IL_i^+ - IL_i^-. \tag{7.4}$$

Note that these are all random variables—they are not decision variables. The decision variables are S_i, the base-stock levels for $i = 1, 2$. We will compute expectations of the random variables once the base-stock levels are set, in order to compute the expected cost.

The complete pooling policy can be stated formally as follows:

(a) If $D_i \le S_i$ for $i = 1, 2$, then $Y_{ij} = Y_{ji} = 0$

(b) If $D_i \ge S_i$ for $i = 1, 2$, then $Y_{ij} = Y_{ji} = 0$

(c) If $D_i < S_i$ and $D_j > S_j$, then

$$Y_{ij} = \min\{S_i - D_i, D_j - S_j\}$$

$$Y_{ji} = 0.$$

This policy is represented graphically in Figure 7.2, which indicates the transshipment quantities and ending inventory levels for all possible realizations of the demand.

7.4.3 Expected Cost

The expected cost per period will be denoted $g(\mathbf{S})$, where $\mathbf{S} = (S_1, S_2)$ is the vector of base-stock levels. $g(\mathbf{S})$ is given by

$$g(\mathbf{S}) = \sum_{i=1}^{2} \left[c_i \mathbb{E}[Q_i] + \sum_{\substack{j=1 \\ j \neq i}}^{2} c_{ij} \mathbb{E}[Y_{ij}] + h_i \mathbb{E}[IL_i^+] + p_i \mathbb{E}[IL_i^-] \right]. \tag{7.5}$$

In order to minimize $g(\mathbf{S})$, we need to compute $\mathbb{E}[Q_i]$, $\mathbb{E}[Y_{ij}]$, $\mathbb{E}[IL_i^+]$, and $\mathbb{E}[IL_i^-]$. First note that

$$\mathbb{E}[Q_i] = S_i - \mathbb{E}[IL_i] \tag{7.6}$$

$$\mathbb{E}[IL_i] = \mathbb{E}[IL_i^+] - \mathbb{E}[IL_i^-]. \tag{7.7}$$

(7.6) follows from the fact that the order quantity is the difference between the target level and the ending inventory in the previous period, while (7.7) follows from (7.4).

The transshipment policy states that $Y_{ij} > 0$ if and only if $D_j > S_j$ and $D_i < S_i$. If this condition holds, the amount shipped is $\min\{S_i - D_i, D_j - S_j\}$. Therefore, we can write

$$\begin{aligned} \mathbb{E}[Y_{ij}] =& \mathbb{E}_{D_i}\left[\mathbb{E}_{D_j}[Y_{ij}|D_i]\right] \\ =& \int_{d_i=0}^{S_i} \left[\int_{d_j=S_j}^{S_i+S_j-d_i} (d_j - S_j) f_j(d_j) dd_j \right. \\ &+ \left. \int_{d_j=S_i+S_j-d_i}^{\infty} (S_i - d_i) f_j(d_j) dd_j \right] f_i(d_i) dd_i. \end{aligned}$$

It can be shown that

$$\mathbb{E}[Y_{ij}] = \int_{d_i=0}^{S_i} F_i(d_i)[1 - F_j(S_i + S_j - d_i)] dd_i. \tag{7.8}$$

Figure 7.2 suggests that the ending inventory level is positive at retailer i if and only if $D_i < S_i$ and $D_j < S_j$ *or* $S_j < D_j \leq S_i + S_j$ and $D_i < S_i + S_j - D_j$. Therefore,

$$\begin{aligned} \mathbb{E}[IL_i^+] =& \int_{d_j=0}^{S_j} \int_{d_i=0}^{S_i} (S_i - d_i) f_i(d_i) f_j(d_j) dd_i dd_j \\ &+ \int_{d_j=S_j}^{S_i+S_j} \int_{d_i=0}^{S_i+S_j-d_j} (S_i + S_j - d_i - d_j) f_i(d_i) f_j(d_j) dd_i dd_j \\ =& \int_{d_i=0}^{S_i} F_i(d_i) F_j(S_i + S_j - d_i) dd_i. \end{aligned} \tag{7.9}$$

Similarly, the ending inventory level is negative at retailer i if and only if $D_i > S_i$ and $D_j > S_j$ *or* $D_j < S_j$ and $D_i > S_i + S_j - D_j$. Therefore,

$$\begin{aligned}
\mathbb{E}[IL_i^-] =& \int_{d_j=S_j}^{\infty} \int_{d_i=S_i}^{\infty} (d_i - S_i) f_i(d_i) f_j(d_j) dd_i dd_j \\
&+ \int_{d_j=0}^{S_j} \int_{d_i=S_i+S_j-d_j}^{\infty} (d_i + d_j - S_i - S_j) f_i(d_i) f_j(d_j) dd_i dd_j \\
=& \mathbb{E}[D_i] - S_i + \int_{d_i=0}^{S_i} F_i(d_i) dd_i - \int_{d_j=0}^{S_j} F_j(d_j) dd_j \\
&+ \int_{d_i=S_i}^{S_i+S_j} F_i(d_i) F_j(S_i + S_j - d_i) dd_i. \qquad (7.10)
\end{aligned}$$

Combining (7.7) with (7.9) and (7.10), we get

$$\begin{aligned}
\mathbb{E}[IL_i] = S_i - \mathbb{E}[D_i] -& \int_{d_i=0}^{S_i} F_i(d_i) dd_i + \int_{d_j=0}^{S_j} F_j(d_j) dd_j \\
&+ \int_{d_i=0}^{S_i} F_i(d_i) F_j(S_i + S_j - d_i) dd_i \\
&- \int_{d_i=S_i}^{S_i+S_j} F_i(d_i) F_j(S_i + S_j - d_i) dd_i. \qquad (7.11)
\end{aligned}$$

This gives us $\mathbb{E}[Q_i]$ using (7.6), so we now have all the components we need to compute $g(\mathbf{S})$. We won't write out $g(\mathbf{S})$ in its entirety since it's a long formula, but it's straightforward to do so using (7.5). As in several of the inventory optimization models we have seen so far, $g(\mathbf{S})$ cannot be optimized in closed form. In other words, we can't set the derivative to 0 and solve for $\mathbf{S}$ in the form $S_1^* =$ [something] and $S_2^* =$ [something]. Instead, we must use numerical methods—general-purpose nonlinear programming algorithms—to solve the problem.

7.4.4 Benefits of Transshipments

Transshipments are beneficial both by reducing costs and by improving service levels. The cost reduction is evident from assumption (a) on page 238—transshipments are less costly than holding and stockouts. Put another way, the transshipment model can be obtained from a "no-transshipment" model by relaxing a constraint—therefore, the optimal cost can only improve (or stay the same).

We will next examine the effect of transshipments on both type-1 and type-2 service levels. (See Section 5.3.1.3 for definitions.) Let

$\alpha_i^0(\mathbf{S})$ = type-1 service level at retailer i if transshipments are not allowed and base-stock levels are set to $\mathbf{S}$

$\alpha_i(\mathbf{S})$ = type-1 service level at retailer i if transshipments are allowed and base-stock levels are set to $\mathbf{S}$

$\beta_i^0(\mathbf{S})$ = type-2 service level at retailer i if transshipments are not allowed and base-stock levels are set to $\mathbf{S}$

$\beta_i(\mathbf{S})$ =type-2 service level at retailer i if transshipments are allowed and base-stock levels are set to $\mathbf{S}$

We will show that transshipments improve both types of service levels. In fact, we will quantify the improvement. We will prove that transshipments improve the service levels for a *given* base-stock level, but this, in turn, implies that the *optimal* solution with transshipments has a higher service level than the optimal solution without transshipments. (Why?)

Theorem 7.2 *Transshipments increase the type-1 service level by the marginal decrease in the expected transshipment quantity for a unit increase in the base-stock level; that is,*

$$\alpha_i(\mathbf{S}) = \alpha_i^0(\mathbf{S}) + \left|\frac{\partial \mathbb{E}[Y_{ji}]}{\partial S_i}\right|$$

for $i = 1, 2$.

Proof. Since $\alpha_i^0(\mathbf{S})$ is the probability that no stockout occurs in a given period with no transshipments,

$$\alpha_i^0(\mathbf{S}) = F_i(S_i). \tag{7.12}$$

Now, no stockouts occur at retailer i in the system with transshipments if either $D_i \le S_i$ *or* $D_i > S_i$ and retailer j has sufficient excess inventory to meet i's excess demand. Therefore,

$$\begin{aligned}
\alpha_i(\mathbf{S}) &= \mathbb{P}(D_i \le S_i) + \mathbb{P}(D_j < S_j \text{ and } S_i < D_i < S_i + S_j - D_j) \\
&= F_i(S_i) + \int_{d_j=0}^{S_j} \left[\int_{d_i=S_i}^{S_i+S_j-d_j} f_i(d_i)dd_i\right] f_j(d_j)dd_j \\
&= F_i(S_i) + \int_{d_j=0}^{S_j} [F_i(S_i+S_j-d_j) - F_i(S_i)] f_j(d_j)dd_j \\
&= F_i(S_i) + \int_{d_j=0}^{S_j} F_i(S_i+S_j-d_j) f_j(d_j)dd_j - F_i(S_i)F_j(S_j)
\end{aligned} \tag{7.13}$$

Differentiating (7.8) with respect to S_i using Leibniz's rule (C.49) gives

$$\frac{\partial \mathbb{E}[Y_{ji}]}{\partial S_i} = F_i(S_i)F_j(S_j) - \int_{d_j=0}^{S_j} F_i(S_i+S_j-d_j) f_j(d_j)dd_j. \tag{7.14}$$

Therefore,

$$\alpha_i = \alpha_i^0 - \frac{\partial \mathbb{E}[Y_{ji}]}{\partial S_i}, \tag{7.15}$$

but since

$$\frac{\partial \mathbb{E}[Y_{ji}]}{\partial S_i} = -\int_{d_j=0}^{S_j} \left[\int_{d_i=S_i}^{S_i+S_j-d_j} f_i(d_i)dd_i\right] f_j(d_j)dd_j < 0,$$

(from (7.13) and (7.14)), we can write (7.15) as

$$\alpha_i = \alpha_i^0 + \left|\frac{\partial \mathbb{E}[Y_{ji}]}{\partial S_i}\right|, \tag{7.16}$$

as desired. ■

Theorem 7.3 *Transshipments increase the type-2 service level at retailer i by the ratio of the expected transshipment quantity from j to i to the expected demand at i; that is,*

$$\beta_i(\mathbf{S}) = \beta_i^0(\mathbf{S}) + \frac{\mathbb{E}[Y_{ji}]}{\mathbb{E}[D_i]}$$

for $i = 1, 2$.

Proof. Omitted; see Problem 7.7. ■

As you might expect, the larger the base-stock levels are, the better the post-transshipment service levels are:

Theorem 7.4 *The type-1 and type-2 service levels (with transshipments) at both i and j are nondecreasing with S_i.*

Proof. Omitted. ■

With more than two retailers, transshipment problems become much harder to analyze. It is often true that a base-stock *replenishment* policy is still optimal in this case (Robinson 1990). In general, it is difficult to determine the optimal *transshipment* policy, so some authors use heuristic policies such as "grouping" policies in which retailers are divided into groups using some logical rules, and then transshipments are allowed only within groups. Models with a small number of retailers, say 3, usually assume complete pooling, even though this policy may not be strictly optimal. Other transshipment policies are possible, of course—for example, Tagaras (1999) compares complete pooling to a random transshipment policy (in which, for example, we choose randomly between two retailers with positive inventory to ship to a retailer with negative inventory) and a risk-balancing policy (which tries to account for the risk of stockout in at least the next period). Lien et al. (2011) propose a "chaining" structure in which all retailers are connected in a single loop; they show that this structure, while suboptimal, outperforms others. Fortunately, it is usually true that a base-stock replenishment policy is optimal even if a nonoptimal transshipment policy is used.

Often, these models are so complex that even the expected cost cannot be calculated using formulas, and instead must be estimated using simulation. In this case, an optimization-by-simulation procedure, such as infinitesimal perturbation analysis (IPA), is used to find the optimal base-stock levels (Herer et al. 2006). One insight to come from these papers is that a small increase in the flexibility with which transshipments are allowed can lead to large decreases in cost. Therefore, more flexible transshipment policies may be preferable, even if they are more difficult to analyze and implement.

7.5 PROCESS FLEXIBILITY

7.5.1 Introduction

Manufacturers in most industries today face increasingly demanding customers and increasingly fierce competition. These factors have led to a huge proliferation in product varieties offered by manufacturers of everything from breakfast cereals to automobiles. For example, the number of car and light truck models for sale in the United States rose from

195 in 1984 to 282 in 2004 (Van Biesebroeck 2007). This so-called *product proliferation* leads to increased diversity and unpredictability of demand. At the same time, firms are under increasing financial pressure to keep capacity as tight as possible, which makes it crucial for manufacturing facilities to have the flexibility to produce a range of products.

The importance of flexibility can be demonstrated by some examples from the automotive industry:

- BMW designs its factory to build cars with the specific colors, features, and options requested by customers. (In contrast, many other auto manufacturers offer a more limited range of combinations, which are ordered by dealers, not by individual customers.) A customer can even change the specifications of his or her car as late as 5 days before the car is built (Henry 2009).

- In 2000–2001, Chrysler saw an unexpectedly large demand for its new PT Cruiser model, while the demand for another car, the Neon, was lower than forecast. As a result, there was a shortage of the PT Cruiser while a manufacturing plant in Belvidere, IL that built only Neons—which have many similar parts as the PT Cruiser—had excess capacity. Chrysler's lack of flexibility to reassign PT Cruiser production to the Belvidere plant cost the company nearly $500 million in lost profit (Biller et al. 2006).

- Learning from this mistake, Chrysler invested heavily in the mid-2000s to ensure that its factories are more flexible and can each make more than one type of vehicle. The Belvidere plant began to make three additional models, and it produced roughly twice as many vehicles in 2006 as it did in 2005. Chrysler Group's CEO, Thomas LaSorda, said that the extra flexibility "gives us a wider margin of error" (Boudette 2006).

- Ford Motor Company invested $485 million to retool two Canadian engine plants with flexible systems. The redesigned plants can produce multiple types of engines and, just as importantly, can switch production from one to another in a matter of hours or days, rather than months. Chris Bolen, the manager of one of the plants, said that "the initial investment is slightly higher, but long-term costs are lower in multiples." The company also had a plan to convert the systems at most of its other engine and transmission plants all over the world to flexible ones (Phelan 2002).

- In the late 1990s, Honda invested $400 million to make its three plants in Ohio flexible. The increased flexibility allowed the company to keep its production closely in line with demand patterns that changed rapidly during the 2000s due to wide fluctuations in gasoline prices and to the global recession. Because most Honda vehicles are designed to be assembled using a similar process, plants can be flexible and can change production from one product to another in as little as five minutes (Linebaugh 2008).

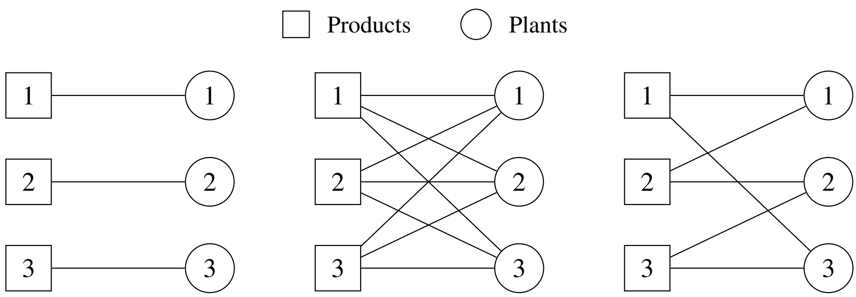
Flexibility can provide a firm with a competitive advantage by allowing it to react quickly to changing demand patterns and supply conditions. It is becoming an increasingly prevalent practice in a wide range of industries, including apparel (DesMarteau 1999) and semiconductors and electronics (McCutcheon 2004). Greater flexibility entails a greater up-front investment, however, and this trade-off must be carefully considered.

In this chapter, we discuss models for evaluating the effectiveness of, and optimizing, *process flexibility*, by which we mean the ability to manufacture a variety of products at the same facility, the ability to manufacture a given product at multiple facilities, or both.

7.5.2 Flexibility Design Guidelines

One of the most important questions in designing a flexible supply chain is, "How much flexibility is enough?" If there is no flexibility, then each plant is assigned to a unique product. If the demand for one product is unexpectedly high while that for another product is low, the firm will stock out of the high-demand product and have excess capacity at the plant that makes the low-demand one. At the other extreme, every plant can produce every product, leaving the firm much better able to reconfigure production in response to demands. Jordan and Graves (1995) describe a simple simulation model that shows that, for a particular set of assumptions, the full-flexibility structure resulted in approximately a 12% increase in sales and capacity utilization. On the other hand, this additional flexibility requires additional capital investments. Is full flexibility really required, or would some in-between strategy be sufficient? As we will see below, it is often possible to choose a partial-flexibility strategy that achieves most of the benefit of the full-flexibility structure with a much smaller resource requirement.

It is common to model process flexibility problems using bipartite graphs (i.e., graphs whose nodes are partitioned into two sets such that no edge has both endpoints in the same set). One set of nodes represents the plants, while the other represents the products. If a plant node and a product node are connected by an edge in the network, then the plant is capable of manufacturing the product. Greater flexibility therefore means more edges in the graph. For example, if there are n plants and n products, then in the dedicated (i.e., no-flexibility) system, there are n edges in the graph, whereas in the full-flexibility system, there are n^2. (See Figures 7.3(a) and 7.3(b).)



(a) Dedicated system. (b) Full flexibility. (c) Chaining structure.

Figure 7.3 Examples of flexibility configurations.

We would like to evaluate the effectiveness of a given flexibility structure (i.e., a given set of edges connecting plants and products). There are many possible ways to define and measure this effectiveness. Typically, we assume that, once the demands for each product in a given period are known, the firm assigns production to the various plants, following the plant–product capabilities implied by the edges and satisfying a fixed capacity constraint at each. One of the most popular ways to measure the effectiveness of a flexibility structure is to evaluate the total *shortfall* (i.e., stockouts) that occurs after the production is optimized and demands are satisfied.

The problem of optimizing production to minimize the shortfall (or, equivalently, maximize the sales) when the demands are known can be formulated as follows (Jordan and Graves 1995). Let $G = (V_1, V_2, E)$ be a bipartite graph consisting of a set V_1 of products, a set V_2 of plants, and an edge set E. Every edge in E has one endpoint in V_1 and one in V_2, indicating a plant–product capability. For example, the full-flexibility structure has edge set $E = \{(i,j)|i \in V_1, j \in V_2\}$. Let d_i be the observed demand realization for product $i \in V_1$, and let C_j be the capacity of plant $j \in V_2$. Let s_i be the shortfall, i.e., the unsatisfied demand, for product i, and let y_{ij} be the number of units of product i produced at plant j, for all $(i,j) \in E$. (s and y are decision variables.) Then, given an observed realization of demand, the production allocation decisions can be optimized, and the minimum total shortfall of a flexibility structure E can be determined by solving the following optimization problem.

$$\text{minimize} \quad \sum_{i \in V_1} s_i \tag{7.17}$$

$$\text{subject to} \quad \sum_{i \in V_1:(i,j) \in E} y_{ij} \le C_j \qquad \forall j \in V_2 \tag{7.18}$$

$$\sum_{j \in V_2:(i,j) \in E} y_{ij} + s_i = d_i \qquad \forall i \in V_1 \tag{7.19}$$

$$y_{ij} \ge 0 \qquad \forall (i,j) \in E \tag{7.20}$$

$$s_i \ge 0 \qquad \forall i \in V_1 \tag{7.21}$$

The objective function (7.17) calculates the total shortfall over all products. (Alternately, we could weight the shortfalls differently, if some products are more important than others.) Constraints (7.18) enforce the capacity restriction at each plant, and constraints (7.19) require the shortfall variable s_i to equal the difference between the demand for product i and the total amount of it produced. Constraints (7.20) and (7.21) are nonnegativity constraints. This problem can be generalized to handle multiechelon supply chains; see Graves and Tomlin (2003) and Chou et al. (2008).

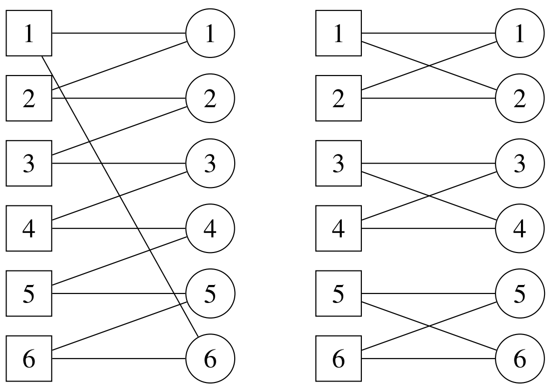
This problem is equivalent to a maximum-flow problem and can therefore be solved efficiently. However, we are interested in evaluating the performance of a given flexibility guideline under *random* demands D_i rather than deterministic demands d_i. (After all, if we knew the demands, we would not need flexibility.) Therefore, we need to solve a stochastic version of the problem, in which we minimize the expected total shortfall over all possible demand realizations. Unfortunately, this problem has a complicated combinatorial and stochastic structure, and finding an optimal solution is challenging. Therefore, researchers have developed intuitive flexibility guidelines that can yield shortfalls that are nearly as low as the shortfall generated by the full-flexibility structure. Moreover, they use far fewer edges and are therefore much less costly to implement. We discuss two of these guidelines next.

Chaining Guideline: Perhaps the best-known flexibility guideline is the *chaining guideline* proposed by Jordan and Graves (1995). (See Figure 7.3(c).) Assume first that $|V_1| = |V_2| = n$. Then the chaining guideline is defined as follows:

- Plant 1 makes products 1 and 2
- Plant 2 makes products 2 and 3
- $\cdots$
- Plant j makes products j and $j + 1$
- $\cdots$
- Plant n makes products n and 1.

This structure uses $2n$ edges. Jordan and Graves (1995) report that chaining can achieve well above 90% of the benefits of the full-flexibility configuration, while using only a fraction of that configuration's n^2 edges. This intuitive result is believed to be true in a wide variety of settings, both analytically and in practice, and has been applied successfully in many industries.

The number of edges is not the only consideration when determining the effectiveness of a chaining guideline. Consider the two flexibility structures in Figure 7.4. Both are chaining structures, both have 12 edges, and in both, every plant makes two products and every product is made at two plants. The structure in Figure 7.4(a) uses a single chain for all products and plants, while that in Figure 7.4(b) partitions the system into three separate chains. The single-chain structure is much more effective, though, achieving nearly twice the benefits (in terms of expected sales) as the three-chain structure in a simulation discussed by Jordan and Graves (1995). (In fact, we prove the optimality of the single-chain structure among all chain-type structures in Section 7.5.3.) The reason is that the single-chain structure allows a greater degree of flexibility in reassigning products to plants than the three-chain structure. For example, if the demand for product 1 is very high and plant 5 has excess capacity, the single-chain structure can take advantage of the discrepancy while the three-chain structure cannot.



(a) One chain. (b) Three chains.

Figure 7.4 Two chaining structures.

Lim et al. (2012) examine the chaining guideline for systems with random supply disruptions that can affect either nodes (representing a disruption of an entire plant) or edges (representing disruptions for particular plant–product pair). For node disruptions, they confirm Jordan and Graves's intuition that longer chains are better, but they find that short chains are preferable when edge failures are the issue.

The discussion so far assumes that the number of products and plants is the same; that the products are identical, as are the plants; and that any plant can be configured to make any product. Real-life situations do not follow this idealized model. Jordan and Graves (1995) outline three guidelines for adding flexibility to chains in more realistic situations:

1. All products should be made by roughly the same number of plants; more precisely, the total capacity of the plants making each product should be roughly the same.

2. All plants should make roughly the same number of products; more precisely, the total expected demand of the products made at each plant should be roughly the same.

3. Longer chains are better than shorter ones.

Node-Expansion Guideline: A more connected guideline is inherently more flexible. With this in mind, Chou et al. (2011) propose the *node-expansion guideline*. The guideline is used to augment a given flexibility structure by adding links iteratively to improve the *node-expansion ratio*. The node-expansion ratio of product $i \in V_1$ is the total capacity of the plants capable of making product i divided by the expected demand for i:

$$\delta_i = \frac{\sum_{j \in V_2:(i,j) \in E} C_j}{\mathbb{E}[D_i]}$$

Similarly, the node-expansion ratio of plant $j \in V_2$ is the total expected demand of the products that can be made at plant j divided by the capacity of plant j:

$$\delta_j = \frac{\sum_{i \in V_1:(i,j) \in E} \mathbb{E}[D_j]}{C_j}$$

Smaller node-expansion ratios suggest products or plants that do not have enough flexibility. The node-expansion guideline says that, at each iteration, we add an edge that is not yet in E in order to increase all node-expansion ratios as much as possible; that is, to increase

$$\delta = \min\left\{\min_{i \in V_1} \delta_i, \min_{j \in V_2} \delta_j\right\}$$

as much as possible. One heuristic for doing this is to add, at each iteration, an edge connecting the product and the plant with the lowest node-expansion ratios, skipping any edges that have already been added. This procedure repeats until the number of edges reaches a predetermined limit.

7.5.3 Optimality of the Chaining Structure

In Section 7.5.2, we remarked that long flexibility chains tend to perform very well, attaining more than 90% of the benefit of a fully connected flexibility graph and performing better than multiple smaller chains (Jordan and Graves 1995). But why is this so? Simchi-Levi and Wei (2012) address that question by proving analytically that a single long chain is optimal among all *2-flexibility designs* for certain types of systems. A 2-flexibility design is one in which each plant can produce exactly two products and each product can be produced by exactly two plants, but the process does not necessarily form a single chain. For example, both structures in Figure 7.4 are 2-flexibility designs. In this section, we will discuss Simchi-Levi and Wei's proof.

We begin with a few definitions. A *balanced* system is one that has an equal number of plants and products and in which all of the plants have the same capacity. In a balanced system of size n (i.e., with n plants and n products), we say the demand D is *exchangeable* if the joint probability distribution of D is the same no matter what order we put the products in.

As before, we will describe a flexibility structure by the set E of edges it contains. We also assign specific notation to certain structures:

- Dedicated design: $\mathcal{D}_n = \{(i,i)|i = 1,2,...,n\}$
- Long-chain design: $\mathcal{C}_n = \mathcal{D}_n \cup \{(i+1,i)|i = 1,2,...,n-1\} \cup \{(n,1)\}$
- Full-flexibility design: $\mathcal{F}_n = \{(i,j)|i,j = 1,2,..,n\}$
- Open chain: $\mathcal{L}_k = \mathcal{D}_k \cup \{(i+1,i)|i = 1,...,k-1\}$, for $k \geq 0$

An open chain $\mathcal{L}_k$ is obtained from a single (closed) chain on nodes $1,\ldots,k$ by removing edge $(1,k)$. We call an edge (i,j) a *dedicated edge* if $i = j$ and a *flexible edge* otherwise.

Formulation (7.17)–(7.21) minimizes the total demand shortfall. It will be more convenient for us to work with an equivalent model that instead maximizes the *performance*, i.e., the sales that result from a particular realization $\mathbf{d}$ of the demand and a given flexibility structure E:

$$P(\mathbf{d},E) = \text{maximize} \quad \sum_{(i,j)\in E} y_{ij} \tag{7.22}$$

$$\text{subject to} \quad \sum_{i\in V_1:(i,j)\in E} y_{ij} \leq C_j \qquad \forall j \in V_2 \tag{7.23}$$

$$\sum_{j\in V_2:(i,j)\in E} y_{ij} \leq d_i \qquad \forall i \in V_1 \tag{7.24}$$

$$y_{ij} \geq 0 \qquad \forall (i,j) \in E \tag{7.25}$$

$$y_{ij} = 0 \qquad \forall (i,j) \notin E \tag{7.26}$$

As in (7.17)–(7.21), here y_{ij} is the number of units of product i produced at plant j. In (7.22)–(7.26), we omit the shortfall variable s and instead maximize the total sales.

Lemma 7.5 *Let E be a flexibility design for a balanced system of size n, with $E \subseteq \mathcal{C}_n$. Let α and β be two flexible edges in E. Then*

$$P(\mathbf{d},E) + P(\mathbf{d},E\setminus\{\alpha,\beta\}) \geq P(\mathbf{d},E\setminus\{\alpha\}) + P(\mathbf{d},E\setminus\{\beta\}). \tag{7.27}$$

Proof. Omitted; see Simchi-Levi and Wei (2012). ■

In other words, Lemma 7.5 says that if we start with E minus two edges, as we add those two edges back into E, we get more marginal benefit from the second edge

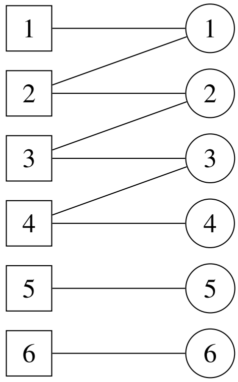


Figure 7.5 $\mathcal{L}_k^n$ structure for $k = 4$ and $n = 6$.

$(P(\mathbf{d}, E) - P(\mathbf{d}, E \setminus \{\beta\}))$ than we did from the first $(P(\mathbf{d}, E \setminus \{\alpha\}) - P(\mathbf{d}, E \setminus \{\alpha, \beta\}))$. Because Lemma 7.5 holds for any demand realization, it must also hold in expectation. For the sake of brevity, for a given edge set E, let $[E] \equiv \mathbb{E}[P(\mathbf{D}, E)]$, where the expectation is over the random demand vector $\mathbf{D}$. Then we have:

Corollary 7.6 *Let E be a flexibility design for a balanced system of size n, with $E \subseteq \mathcal{C}_n$. Let α and β be two flexible edges in E. Then*

$$[E] + [E \setminus \{\alpha, \beta\}] \geq [E \setminus \{\alpha\}] + [E \setminus \{\beta\}]. \tag{7.28}$$

Therefore, any two flexible arcs in the long-chain design *complement* each other, in the sense that having one flexible edge in the system increases the marginal benefit that can be gained when another flexible edge is added.

Our goal is to prove that among all 2-flexibility designs, the long-chain structure maximizes the expected performance, $\mathbb{E}[P(\mathbf{D}, E)]$. We will do that by first showing that as we add edges to the dedicated system to build up to a long chain, each new edge brings more benefit than the previous edge did. Next, we will express the expected performance of the long chain in terms of open chains, and finally, we will prove the optimality of long chains.

Define $\mathcal{L}_1^n = \mathcal{D}_n$ and $\mathcal{L}_k^n = \mathcal{L}_k \cup \{(i, i) | i = k+1, \ldots, n\}$ for $2 \leq k \leq n$. In words, $\mathcal{L}_k^n$ consists of the open chain from plant 1 through product k, plus the dedicated edges between product/plant pairs $k + 1, \ldots, n$. (See Figure 7.5.) Note that $\mathcal{L}_n^n$ is simply $\mathcal{L}_n$. The next lemma shows that the incremental benefit of each additional flexible edge is nondecreasing as the long chain is constructed.

Lemma 7.7 *For any balanced system of size n with exchangeable demand,*

$$\begin{aligned}
[\mathcal{L}_2^n] - [\mathcal{L}_1^n] &\leq [\mathcal{L}_3^n] - [\mathcal{L}_2^n] \\
&\vdots \\
&\leq [\mathcal{L}_n^n] - [\mathcal{L}_{n-1}^n] \\
&\leq [\mathcal{C}_n] - [\mathcal{L}_n^n].
\end{aligned}$$

Proof. For any fixed $2 \le k \le n-1$, let $\alpha = (2,1)$ and $\beta = (k+1,k)$. Note that $\alpha, \beta \in \mathcal{L}^n_{k+1}$. By Corollary 7.6 (treating E as $\mathcal{L}^n_{k+1}$), we have

$$[\mathcal{L}^n_{k+1}] + [\mathcal{L}^n_{k+1} \setminus \{\alpha,\beta\}] \ge [\mathcal{L}^n_{k+1} \setminus \{\alpha\}] + [\mathcal{L}^n_{k+1} \setminus \{\beta\}]. \tag{7.29}$$

Now, if we remove β from $\mathcal{L}^n_{k+1}$, we simply get $\mathcal{L}^n_k$. If we remove α, then by rearranging the product/node pairs so that pair 1 moves to the end (which we are allowed to do since the demands are exchangeable), we again obtain $\mathcal{L}^n_k$. Similarly, if we remove both α and β, we obtain $\mathcal{L}^n_{k-1}$. Therefore, (7.29) becomes

$$[\mathcal{L}^n_{k+1}] + [\mathcal{L}^n_{k-1}] \ge [\mathcal{L}^n_k] + [\mathcal{L}^n_k],$$

or

$$[\mathcal{L}^n_k] - [\mathcal{L}^n_{k-1}] \le [\mathcal{L}^n_{k+1}] - [\mathcal{L}^n_k].$$

Since this holds for all $k = 2, \dots, n-1$, we have now proven all of the inequalities in the lemma except the final one.

To prove the final inequality, let $\alpha = (2,1)$ and $\beta = (1,n)$. Using similar logic as above, we have $\mathcal{C}_n \setminus \{\alpha\} = \mathcal{C}_n \setminus \{\beta\} = \mathcal{L}^n_n$ and $\mathcal{C}_n \setminus \{\alpha,\beta\} = \mathcal{L}^n_{n-1}$. Therefore, by Corollary 7.6 (treating E as $\mathcal{C}_n$), we have

$$[\mathcal{C}_n] + [\mathcal{L}^n_{n-1}] \ge [\mathcal{L}^n_n] + [\mathcal{L}^n_n],$$

or

$$[\mathcal{L}^n_n] - [\mathcal{L}^n_{n-1}] \le [\mathcal{C}_n] - [\mathcal{L}^n_n],$$

completing the proof. ■

The result in Lemma 7.5 holds for any demand realization, which allowed us to prove the same result in expectation in Corollary 7.6. In contrast, Lemma 7.7 holds in expectation, but the same result does not hold for every individual demand instance. (See Problem 7.15.)

Next, we characterize the performance of the long-chain design using the performance of open chains.

Lemma 7.8 *For any balanced system of size n with exchangeable demand, we have*

$$[\mathcal{C}_n] = n([\mathcal{L}_n] - [\mathcal{L}_{n-1}]).$$

Proof. For any demand realization $\mathbf{d}$, one can show (Simchi-Levi and Wei 2012, Theorem 3) that

$$P(\mathbf{d},\mathcal{C}_n) = \sum_{i=1}^{n} (P(\mathbf{d},\mathcal{C}_n \setminus \{(i+1,i)\}) - P(\mathbf{d},\mathcal{C}_n \setminus \{(i,i-1),(i,i),(i+1,i)\})). \tag{7.30}$$

For any $1 \le i \le n$,

$$\mathcal{C}_n \setminus \{(i+1,i)\} = \mathcal{L}_n$$
$$\mathcal{C}_n \setminus \{(i,i-1),(i,i),(i+1,i)\} = \mathcal{L}_{n-1}$$

since the demand is exchangeable. (Imagine removing one diagonal edge, or two consecutive diagonal edges and the horizontal edge between them, and then rearranging the nodes to obtain an open chain.) Taking the expectation of both sides of (7.30), we obtain the desired result. ■

Lemma 7.8 expresses the performance of a long-chain design in terms of the performance of open chains, which are much easier to compute and analyze using a greedy heuristic (see Chou et al. 2010b).

Finally, we are ready to prove the optimality of the single-chain structure (i.e., the long-chain design) among all 2-flexibility designs.

Theorem 7.9 *Let $\mathbb{F}_2$ be the set of all 2-flexibility designs of the system. Then,*

$$\mathcal{C}_n \in \underset{A \in \mathbb{F}_2}{\operatorname{argmax}} \{[A]\}.$$

Proof. Let $A \in \mathbb{F}_2$ be any 2-flexibility design. It suffices to show that $[A] \le [\mathcal{C}_n]$.

By the definition of a 2-flexibility design, A must consist of one or more closed chains. If it consists of only a single chain, then $A = \mathcal{C}_n$, so $[A] \le [\mathcal{C}_n]$ trivially. Suppose instead that A consists of $m \ge 2$ disjoint closed chains, and let n_j be the number of products and plants in the jth closed chain, for $1 \le j \le m$. Then:

$$[A] = \sum_{j=1}^{m} [\mathcal{C}_{n_j}] = \sum_{j=1}^{m} n_j([\mathcal{L}_{n_j}] - [\mathcal{L}_{n_j-1}]).$$

The first equality follows from the fact that A consists of m chains, each of length n_j. The second equality follows from Lemma 7.8. Since $\mathcal{L}^n_{n_j}$ equals $\mathcal{L}_{n_j}$ plus $n - n_j$ disjoint edges and $\mathcal{L}^n_{n_j-1}$ equals $\mathcal{L}_{n_j-1}$ plus $n - n_j + 1$ disjoint edges,

$$[\mathcal{L}_{n_j}] - [\mathcal{L}_{n_j-1}] = [\mathcal{L}^n_{n_j}] - [\mathcal{L}^n_{n_j-1}] + \mathbb{E}[\min\{C_1, D_1\}], \tag{7.31}$$

where $\mathbb{E}[\min\{C_1, D_1\}]$ is the expected sales of the "extra" edge in $\mathcal{L}^n_{n_j-1}$ (which we assume is for plant and product 1, without loss of generality due to the exchangeable demand). Therefore,

$$\begin{aligned}
[A] &= \sum_{j=1}^{m} n_j([\mathcal{L}^n_{n_j}] - [\mathcal{L}^n_{n_j-1}] + \mathbb{E}[\min\{1, D_1\}]) \\
&\le \sum_{j=1}^{m} n_j([\mathcal{L}^n_{n}] - [\mathcal{L}^n_{n-1}] + \mathbb{E}[\min\{1, D_1\}]) \\
&= \sum_{j=1}^{m} n_j([\mathcal{L}_{n}] - [\mathcal{L}_{n-1}]) \\
&= n([\mathcal{L}_{n}] - [\mathcal{L}_{n-1}]) \\
&= [\mathcal{C}_n],
\end{aligned}$$

where the inequality follows from Lemma 7.7, the second equality follows the same logic as (7.31), the third follows from the fact that $\sum_{j=1}^{m} n_j = n$, and the last follows from Lemma 7.8. ■

7.6 A PROCESS FLEXIBILITY OPTIMIZATION MODEL

So far we have discussed flexibility guidelines for symmetric networks, in which all plants have the same capacity and all products have independent, identical demand distributions. However, real systems are much more complex. Jordan and Graves's (1995) three rules of thumb listed on page 248 provide some guidance, but it would be helpful to have a more rigorous, optimization-based approach to design flexibility structures. In addition, the models we have discussed so far ignore the possibility that the investment and operating costs of different flexible resources can be different. For example, it is generally cheaper for a plant to produce two similar products than two very different products.

In addition, some flexible plants are designed for one primary product (or product family), and when it is called upon to produce a different product, production costs may increase—for example, due to additional costs for training workers to produce the new product, or to the change-over time required to switch products on an assembly line. These "recourse" costs are ignored in many process flexibility models. One exception is Chou et al. (2010a), who assume that it costs more for a plant to manufacture products other than those it is primarily designed for. Their results show that chaining can be less beneficial relative to full flexibility when recourse costs are taken into consideration, but that chaining still yields significant benefits over the no-flexibility structure.

Another paper that accounts for recourse costs, as well as nonhomogeneous products and plants, is that of Mak and Shen (2009), which *optimizes* the flexibility structure to maximize the firm's expected profit, accounting for the costs to invest in process flexibility. We discuss their model in this section.

7.6.1 Formulation

As in earlier parts of this chapter, we consider a set V_1 of products, indexed by i, and a set V_2 of plants, indexed by j, each with n elements.[1] Demands for the products are random.

This is a two-stage stochastic optimization model. In the first stage, we decide which edges $(i, j) \in E$ to construct, i.e., which plants should be made capable of producing which products. There is a fixed investment cost of a_{ij} to add edge (i, j), representing the cost of retooling the manufacturing process or purchasing a flexible technology. At the beginning of the second stage, we observe the random demands and then choose production levels for each product at each plant, subject to the flexibility structure chosen in the first stage. There is a production cost of c_{ij} for each unit of plant j's capacity that is used to produce product i and a selling price of p_i for each unit of product i sold. The objective is to maximize the profit, which equals the sales revenue minus the costs of production and flexibility investments.

We model the random product demands using scenarios: The demand for product i in scenario s is given by d_{is}, and the probability that scenario s occurs is q_s.[2]

We summarize the notation as follows:

Sets

[1]To be consistent with the literature, we assume that $|V_1| = |V_2| = n$. However, it is trivial to allow these numbers to be different; see Mak and Shen (2009).

[2]Mak and Shen (2009) consider a much more general multivariate demand model. We consider the scenario-based approach here for the sake of simplicity.

V_1 = set of products
V_2 = set of plants
S = set of scenarios

Parameters

a_{ij} = cost to invest in technology that allows plant j to produce product i
c_{ij} = cost to produce one unit of product i at plant j
p_i = revenue from selling one unit of product i
C_j = capacity of plant j
d_{is} = demand for product i in scenario s
q_s = probability that scenario s occurs

Decision Variables

x_{ij} = 1 if plant j is configured to produce product i, 0 otherwise
y_{ijs} = the number of units of product i produced at plant j in scenario s

We formulate the model for optimizing process flexibility as follows:

$$\text{maximize} \quad \sum_{i\in V_1}\sum_{j\in V_2}\left[-a_{ij}x_{ij} + \sum_{s\in S} q_s(p_i - c_{ij})y_{ijs}\right] \tag{7.32}$$

$$\text{subject to} \quad \sum_{i\in V_1} y_{ijs} \le C_j \qquad \forall j \in V_2, \forall s \in S \tag{7.33}$$

$$\sum_{j\in V_2} y_{ijs} \le d_{is} \qquad \forall i \in V_1, \forall s \in S \tag{7.34}$$

$$y_{ijs} \le d_{is}x_{ij} \qquad \forall i \in V_1, \forall j \in V_2, \forall s \in S \tag{7.35}$$

$$x_{ij} \in \{0,1\} \qquad \forall i \in V_1, \forall j \in V_2 \tag{7.36}$$

$$y_{ijs} \ge 0 \qquad \forall i \in V_1, \forall j \in V_2, \forall s \in S \tag{7.37}$$

The objective function (7.32) calculates the expected profit—the expected sales revenue minus investment costs and expected production costs. Constraints (7.33) enforce the capacity limit at each plant in each scenario. Constraints (7.34) require the amount of product i produced in scenario s to be less than or equal to the demand. Without these constraints, the model might choose to produce more than the demand in order to increase the profit. Note, however, that the formulation does not require the demand to be met in full. A product's demand may not be met in full, or at all, if there is insufficient capacity or if it is not profitable to meet the demand. Constraints (7.35) allow production of product i at plant j in scenario s only if that capability was established in the first stage. Constraints (7.36) and (7.37) require the x variables to be binary and the y variables to be nonnegative.

The second stage of this problem (i.e., the problem in the y variables) is similar to the deterministic model (7.17)–(7.21) except that (1) the goal is to maximize profit rather than minimize shortfall and (2) the plant–product capabilities are first-stage decisions rather than exogenous factors.

7.6.2 Lagrangian Relaxation

We now describe a Lagrangian relaxation algorithm to solve the process flexibility design model. (Lagrangian relaxation is covered in more detail in Section 8.2.3 and in

Appendix D.1.) We relax constraints (7.34) and (7.35) with Lagrange multipliers τ and η, respectively. Since we are relaxing $\leq$ constraints in a maximization problem, τ and η are both restricted to be nonnegative (see Section D.1.5). The Lagrangian subproblem becomes:

$$
\begin{aligned}
\text{maximize} \quad & \sum_{i \in V_1} \sum_{j \in V_2} \left[-a_{ij} x_{ij} + \sum_{s \in S} q_s (p_i - c_{ij}) y_{ijs} \right] \\
& + \sum_{i \in V_1} \sum_{s \in S} \left[\tau_{is} \left(d_{is} - \sum_{j \in V_2} y_{ijs} \right) + \sum_{j \in V_2} \eta_{ijs} \left(d_{is} x_{ij} - y_{ijs} \right) \right] \\
= & \sum_{i \in V_1} \sum_{j \in V_2} \left(-a_{ij} + \sum_{s \in S} \eta_{ijs} d_{is} \right) x_{ij} \\
& + \sum_{i \in V_1} \sum_{j \in V_2} \sum_{s \in S} [q_s (p_i - c_{ij}) - \tau_{is} - \eta_{ijs}] y_{ijs} + \sum_{i \in V_1} \sum_{s \in S} \tau_{is} d_{is} \qquad (7.38)
\end{aligned}
$$

$$
\begin{aligned}
\text{subject to} \quad & \sum_{i \in V_1} y_{ijs} \leq C_j && \forall j \in V_2, \forall s \in S && (7.39) \\
& x_{ij} \in \{0, 1\} && \forall i \in V_1, \forall j \in V_2 && (7.40) \\
& y_{ijs} \geq 0 && \forall i \in V_1, \forall j \in V_2, \forall s \in S && (7.41)
\end{aligned}
$$

This problem decouples into two subproblems, one involving only x and one involving only y. The x-problem is trivial to solve: We simply set $x_{ij} = 1$ if

$$-a_{ij} + \sum_{s \in S} \eta_{ijs} d_{is} > 0$$

and set $x_{ij} = 0$ otherwise. Solving the y-problem amounts to solving the following problem for each j and s:

$$
\begin{aligned}
(\mathrm{P}_{js}) \quad \text{maximize} \quad & \sum_{i \in V_1} a_i y_i && (7.42) \\
\text{subject to} \quad & \sum_{i \in V_1} y_i \leq C_j && (7.43) \\
& y_i \geq 0 \qquad \forall i \in V_1 && (7.44)
\end{aligned}
$$

where

$$
\begin{aligned}
a_i &= q_s (p_i - c_{ij}) - \tau_{is} - \eta_{ijs} \\
y_i &= y_{ijs}.
\end{aligned}
$$

This problem, too, is easy: We simply set $y_i = C_j$ for the i that has the largest a_i and $y_i = 0$ for all other i. (If $a_i \leq 0$ for all i, then we set $y_i = 0$ for all i.) The problem could be strengthened somewhat by adding a constraint

$$y_{ijs} \leq d_{is} \quad \forall i \in V_1, \forall j \in V_2, \forall s \in S$$

to the original problem. This constraint is redundant in the original problem but strengthens the y-problem by reducing its optimal objective value (or leaving it the same), thereby

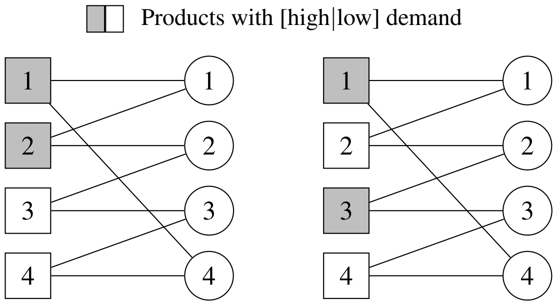


Figure 7.6 Examples of different chaining structures for nonhomogeneous demand case.

tightening the Lagrangian upper bound. If we do this, the y-problem becomes a continuous knapsack problem, which is still easy to solve.

In Mak and Shen's (2009) formulation of this problem, the demands are modeled using a continuous, multivariate distribution, rather than the discrete scenarios used here. In effect, this means that there are an infinite number of demand scenarios, and hence, we must relax an infinite number of constraints of type (7.34) and (7.35). To handle this issue, Mak and Shen propose the use of scenario-independent Lagrange multipliers; that is, to omit the subscript s from τ and η and to use the same multipliers for all scenarios. This results in a weaker upper bound from the Lagrangian subproblem than if the multipliers depend on the scenario, but it also leads to a more tractable Lagrangian dual problem. In general, the quality of the bound is better if the demand variability is relatively small. (See, for example, Kunnumkal and Topaloglu (2008) for a discussion.) This approach has been used successfully in stochastic network flow and stochastic dynamic programming problems (Cheung and Powell 1996, Topaloglu 2009).

Feasible solutions to the original problem can be obtained from solutions to the Lagrangian subproblem in order to obtain lower bounds. To do this, we set the first-stage (x) variables to their values from the subproblem. Once these variables are fixed, the y variables can be determined by solving a network flow problem for each scenario s. (For the continuous-demand case in Mak and Shen (2009), the y variables must be determined by solving a stochastic linear program.)

Mak and Shen (2009) compare the solutions obtained from this flexibility optimization model with the simple chaining structure. When the products are identical in terms of demand distribution and production cost, the two approaches produce solutions with similar expected profit. For nonhomogenous products, the performance of the chaining strategy can be sensitive to the sequences of the products and plants. For example, if there are two high-demand products and two low-demand products, then the solutions will be different if we number the high-demand products as $i = 1, 2$ than if we number them as $i = 1, 3$. (See Figure 7.6.) Therefore, the performance of the straightforward chaining structure, in which plant j produces products $i = j$ and $j + 1$, may depend on how the products happen to be indexed. On the other hand, the process flexibility design model discussed in this section accounts for these nonhomogeneities explicitly. As a result, this approach outperforms the simple chaining approach considerably for some problem instances.

CASE STUDY 7.1 Risk Pooling and Inventory Management at Yedioth Group

Yedioth Group is the largest media group in Israel. The group sells magazines and other publications through thousands of independent retailers, which receive merchandise at the beginning of each week (for weekly magazines) or each day (for daily newspapers). The retailers cannot order additional stock if they run out during the period, and unused inventory is collected at the end of the period and scrapped. Therefore, the retailers function exactly as in the newsvendor problem (in addition to being actual newsvendors).

Demand for magazines and newspapers is typically highly variable. At the same time, the lost-sales penalty is considered to be very high because unmet demands mean lost advertising revenue. Yedioth reimburses the retailers in full for any unused inventory, which means the retailers incur no overage risk, and therefore, their optimal base-stock level would be infinite. Therefore, Yedioth chooses delivery quantities directly for the retailers, based on demand forecasts. Inventory is stored at each of the retailers, and the retailers cannot transship to one another. Therefore, under this setting, there is no risk pooling, either physical pooling or information pooling.

Yedioth suspected this was not an ideal distribution strategy, so they partnered with researchers from the Technion—Israel Institute of Technology and from the Massachusetts Institute of Technology (MIT) to develop a better approach. Their research is described by Avrahami et al. (2014).

The key idea behind the new approach is to make two deliveries per week. Demand during the first part of the week is used to choose delivery quantities for the second delivery, which can be used to restock the retailers whose inventory is low. Groups of about 80–100 retailers are each handled by a single sales agent, and their model makes decisions for one retailer group at a time. The model makes three sets of decisions: (1) how many copies to print at the beginning of the week (printing more copies during the week is prohibitively expensive, so there is only one print run); (2) how many copies to deliver to each retailer in the group at the beginning of the week; and (3) how many copies to deliver to the retailers in the middle of the week. The first two decisions must be made before observing any demand information for the week, while the third decision can exploit the observed demands in the first part of the week.

Avrahami et al. (2014) formulate this problem as a two-stage stochastic optimization model that makes decisions for the n retailers assigned to a given group. The model has two subperiods, denoted $t = 1, 2$; subperiods 1 and 2 correspond to the portions of the week before and after the second delivery, respectively. D_i^t is a random variable representing the demand at retailer i in subperiod t; its distribution is estimated using historical data. The relevant costs are c, the production cost per unit; h, the overage cost per unit not sold by the end of subperiod 2 (h represents a disposal cost, i.e., a negative salvage value); and p, the stockout cost per unit of unmet demand in either subperiod. There is no distribution cost.

The model has the following decision variables: y_i^t is the on-hand inventory at retailer i after items are delivered in subperiod t; Q^1 is the number of units not delivered in subperiod 1; and x_i is the inventory level (positive or negative) at retailer i at the end of subperiod 1.

At the beginning of the week, we must choose y_i^1 and Q^1, accounting for the random demand in the first subperiod and the subsequent (random) deliveries that will be made in the second subperiod. That is, we wish to solve

$$\begin{aligned} \text{minimize} \quad & c\left(\sum_{i=1}^{n} y_i^1 + Q^1\right) + \mathbb{E}\left[P_2(Q^1, y_1^1 - D_1^1, \ldots, y_n^1 - D_n^1)\right] && (7.45) \\ \text{subject to} \quad & y_i^1 \geq 0 \qquad \forall i = 1, \ldots, n && (7.46) \\ & Q^1 \geq 0, && (7.47) \end{aligned}$$

where $P_2(Q^1, x_1, \ldots, x_n)$ is the expected cost in subperiod 2, given that subperiod 1 ends with Q^1 undelivered units and an inventory level of x_i at retailer i. That is:

$$\begin{aligned} P_2(Q^1, x_1, \ldots, x_n) = \text{minimize} \quad & \mathbb{E}\left[\sum_{i=1}^{n} \left[h(y_i^2 - D_i^2)^+ + p(D_i^2 - y_i^2)^+\right]\right] && (7.48) \\ \text{subject to} \quad & y_i^2 = x_i + Q_i^2 \qquad \forall i = 1, \ldots, n && (7.49) \\ & \sum_{i=1}^{n} Q_i^2 = Q^1 && (7.50) \\ & Q_i^2 \geq 0 \qquad \forall i = 1, \ldots, n && (7.51) \end{aligned}$$

Note that the stockout cost is not incurred until the end of the second subperiod, since first-subperiod stockouts are passed along to the second subperiod via the term $y_i^1 - D_i^1$.

The expectations are taken over the first- and second-subperiod demands (respectively), whose distributions are discretized so that the expectations are sums rather than integrals. This allows the resulting objective functions to be linearized, but it also makes the scenario space huge; for example, if there are 50 retailers and each can have high, medium, or low demand in each subperiod, we have $(3^{50})^2 \approx 5.2 \times 10^{47}$ scenarios. Therefore, Avrahami et al. (2014) use sampling to estimate the expectations. They show that the objective function is convex and use this property to develop a stochastic subgradient-based optimization algorithm. The algorithm executes very quickly, solving each retailer group in a few seconds.

This model looks very different from the risk-pooling models discussed in Section 7.2, but the principle is very similar. By adding a second delivery in each week, Yedioth can maintain a centralized inventory for part of the week, which allows it to exploit the risk-pooling effect across the retailers. Alternately, we can interpret this as a postponement strategy in which the differentiation refers to the delivery to retailers (rather than customization of the product).

Yedioth initially implemented the new approach for only one product, the weekly magazine *La'Isha*, and for only 50 retailers. This pilot project was very successful, so the company expanded it to more publications and many more retailers. The initial results showed a 9% reduction in production levels (without a decrease in sales) and a 35% reduction in product returns. The company estimates savings of $1,000,000 per year in printing costs when the new approach is rolled out to all 8,000 of its retailers.

PROBLEMS

7.1 (Risk-Pooling Example) Three distribution centers (DCs) each face normally distributed demands, with $D_1 \sim N(22, 8^2)$, $D_2 \sim N(19, 4^2)$, and $D_3 \sim N(17, 3^2)$. All three DCs have a holding cost of $h = 1$ and $p = 15$, and all three follow a periodic-review base-stock policy using their optimal base-stock levels.

- **a)** Calculate the expected cost of the decentralized system.
- **b)** Suppose demands are uncorrelated among the three DCs: $\rho_{12} = \rho_{13} = \rho_{23} = 0$. Calculate the expected cost of the centralized system.
- **c)** Suppose $\rho_{12} = \rho_{13} = \rho_{23} = 0.75$. Calculate the expected cost of the centralized system.
- **d)** Suppose $\rho_{12} = 0.75$, $\rho_{13} = \rho_{23} = -0.75$. Calculate the expected cost of the centralized system.

7.2 (No Soup for You) A certain New York City soup vendor sells 15 varieties of soup. The number of customers who come to the soup store on a given day has a Poisson distribution with a mean of 250. A given customer has an equal probability of choosing each of the 15 varieties of soup, and if his or her chosen variety of soup is out of stock (no pun intended), he or she will leave without buying any soup.

You may assume (although it is not necessarily a good assumption) that the demands for different varieties of soup are independent; that is, if the demand for variety i is high on a given day, that doesn't indicate anything about the demand for variety j.

Every type of soup sells for \$5 per bowl, and the ingredients for each bowl of soup cost the soup vendor \$1. Any soups (or ingredients) that are unsold at the end of the day must be thrown away.

- **a)** How many ingredients of each variety of soup should the soup vendor buy? What is the restaurant's total expected underage and overage cost for the day?
- **b)** What is the probability that the vendor stocks out of a given variety of soup?
- **c)** Now suppose that the soup vendor wishes to streamline his offerings by reducing the selection to 8 varieties of soup. Assume that the total demand distribution does not change, but now the total demand is divided among 8 soup varieties instead of 15. As before, assume that a customer finding his or her choice of soup unavailable will leave without purchasing anything. Now how many ingredients of each variety of soup should the vendor buy? What is the restaurant's total expected underage and overage cost for the day?
- **d)** In a short paragraph, explain how this problem relates to risk pooling.

Note: You may use the normal approximation to the Poisson distribution, but make sure to specify the parameters you are using.

7.3 (In-Flight Trash) On a certain airline, the flight attendants collect trash during flights and deposit it all into a single receptacle. Airline management is thinking about instituting an on-board recycling program in which waste would be divided by the flight attendants and placed into three separate receptacles: one for paper, one for cans and bottles, and one for other trash.

The volume of each of the three types of waste on a given flight is normally distributed. The airline would maintain a sufficient amount of trash-receptacle space on each flight so that the probability that a given receptacle becomes full under the new system is the same as the probability that the single receptacle becomes full under the old system.

Would the new policy require the same amount of space, more space, or less space for trash storage on each flight? Explain your answer in a short paragraph.

7.4 (Days-of-Supply Policies) Rather than setting safety stock levels using base-stock or (r, Q) policies, some companies set their safety stock by requiring a certain number of "days of supply" to be on hand at any given time. For example, if the daily demand has a mean of 100 units, the company might aim to keep an extra 7 days of supply, or 700 units, in inventory. This policy uses μ instead of σ to set safety stock levels.

Consider the N-DC system described in Section 7.2.1, with independent demands across DCs ($\rho_{ij} = 0$ for $i \neq j$). You may assume that all DCs are identical: $\mu_i = \mu$ and $\sigma_i = \sigma$ for all i. Assume that μ and σ refer to *weekly* demands, and that orders are placed by the DCs once per week. Finally, assume that each DC follows a days-of-supply policy with k days of supply required to be on hand as safety stock; each DC's order-up-to level is then

$$S = \mu + \frac{k}{7}\mu.$$

a) Prove that the centralized and decentralized systems have the *same amount* of total inventory.

b) Derive expressions for g_D^* and g_C^*, the total expected costs of the decentralized and centralized systems. Your expressions may *not* involve integrals; they *may* involve the standard normal loss function, $\mathscr{L}(\cdot)$.

Hint: Since the DCs are not following the optimal stocking policy, the cost is analogous to (4.29), not to (4.30).

c) Prove that $g_C^* < g_D^*$.

d) Explain in words how to reconcile parts (a) and (c)—how can the centralized cost be smaller even though the two systems have the same amount of inventory?

7.5 (Negative Safety Stock) Consider the N-DC system described in Section 7.2.1, with independent demands across DCs ($\rho_{ij} = 0$). Suppose that the holding cost is greater than the stockout cost: $h > p$.

a) Prove that *negative* safety stock is required at DC i—that the base-stock level is less than the mean demand.

b) Prove that the total inventory (cycle stock and safety stock) required in the decentralized system (each DC operating independently) is *less* than the total inventory required in the centralized system (all DCs pooled into one). (This is the opposite of the result in Section 7.2.)

c) Prove that, despite the result from part (b), the total expected cost of the centralized system is less than that of the decentralized system ($g_C^* < g_D^*$).

d) Explain in words how to reconcile parts (b) and (c)—how can it be less expensive to hold more inventory?

7.6 (Rationalizing DVR Models) A certain brand of digital video recorder (DVR) is available in three models, one that holds 40 hours of TV programming, one that holds 80 hours, and one that holds 120 hours. The lifecycle for a given DVR model is short, roughly 1 year. Because of long manufacturing lead times, the company must manufacture all of

Table 7.2 DVR parameters for Problem 7.6.

Storage Space	Manufacturing Cost (c_i)	Selling Price (r_i)	Goodwill Cost (g_i)	Mean Annual Demand (μ_i)	SD of Annual Demand (σ_i)
40	80	120	150	40,000	12,000
80	90	150	150	55,000	15,000
120	100	250	150	25,000	8,000

the units it intends to sell before the DVRs go on the market, and it will not have another opportunity to manufacture more before the end of the products' 1-year life cycles.

Demand for DVRs is highly volatile, and customers are very picky. A customer who wants a given model but finds that it's out of stock will almost never change to a different model—instead, he or she will buy a competitor's product. In this case, the firm incurs both the lost profit and a loss-of-goodwill cost. Moreover, any DVRs that are unsold at the end of the year are taken off the market and destroyed, with no salvage value (or cost).

The cost, revenue, and demand parameters for the three models of DVR are given in Table 7.2. Demands are normally distributed with the parameters specified in the table. Moreover, demands for the 80- and 120-hour models are negatively correlated, with a correlation coefficient of $\rho_{80,120} = -0.4$. (Demands for the 40-hour model are independent of those for the other two models.)

The company is currently designing its three models for next year, and a very smart supply chain manager noticed that although the models sell for different prices, they cost nearly the same amount to manufacture. The manager thus proposed that the firm manufacture only a *single* model, containing 120 hours of storage space. When customers purchase a DVR, they specify how much storage space they'd like it to have (either 40, 80, or 120 hours) and pay the corresponding price, and the unit is activated with that much space. If the customer asks for 40 or 80 hours, the remaining storage space simply goes unused. This change can be made with software rather than hardware and therefore costs very little to make.

a) Let Q_i be the quantity of model i manufactured, $i = 1, \ldots, 3$, if the supply chain manager's proposal is *not* followed. Write the firm's expected profit for model i as a function of Q_i.

b) Find the optimal order quantities Q_i^* and the corresponding total optimal expected profit (for all three models).

c) Let Q be the quantity of the single model manufactured if the manager's proposal *is* followed. Write the firm's total expected profit as a function of Q. Although it is not entirely accurate to do so, you may assume that the expected selling price for the single model is given by a weighted average of the r_i, with weights given by the μ_i.

d) Find the optimal order quantity Q and the corresponding optimal expected profit. Based on this analysis, should the firm follow the manager's suggestion?

e) What other factors should the firm consider before deciding whether to implement the manager's proposal?

7.7 (Proof of Theorem 7.3) Prove Theorem 7.3.

7.8 (Transhipment Simulation) Build a spreadsheet simulation model for the two-retailer transshipment problem from Section 7.4. Your spreadsheet should include columns for the demand at each location; the inventory at each location at the start of the period, before transshipments, and after transshipments; the amount transshipped; and the costs for the period. Assume that demands are Poisson with mean λ_i per period and that

$$\begin{aligned} \lambda_1 &= 30 & \lambda_2 &= 20 \\ c_1 &= 1.2 & c_2 &= 1.7 \\ h_1 &= 0.6 & h_2 &= 0.8 \\ p_1 &= 8.0 & p_2 &= 8.0 \\ c_{12} &= 3.0 & c_{21} &= 3.0. \end{aligned}$$

Use $S_1 = 33$ and $S_2 = 22$ as the base-stock levels, and assume that both retailers begin the simulation with $S_i - \lambda_i$ units on-hand (that is, at the start of period 1, retailer i needs to order λ_i units to bring its inventory position to S_i).

a) Simulate the system for 500 periods and include the first 10 rows of your spreadsheet in your report.

b) Compute the average ordering, transshipment, holding, and penalty costs per period from your simulation.

c) Compute the expected transshipment quantity from retailer 1 to retailer 2 ($\mathbb{E}[Y_{12}]$) and the expected ending inventory at retailer 1 ($\mathbb{E}[IL_1^+]$) using (7.8) and (7.9). To compute these quantities, you will need to evaluate some integrals numerically.

d) Compare the results from parts (a) and (c). How closely do the simulated and actual quantities match?

e) By trial and error, try to find the values of S_1 and S_2 that minimize the simulated cost. What are the optimal values, and what is the optimal expected cost?

7.9 (Binary Transshipments) Consider the transshipment model from Section 7.4, except now suppose the demands are binary. That is, the demands can only equal 0 or 1, and they are governed by a Bernoulli distribution: $D_i = 1$ with probability q_i and $D_i = 0$ with probability $1 - q_i$, for $i = 1, 2$. All of the remaining assumptions from Section 7.4.2 hold.

Your goal in this problem will be to formulate the expected cost and evaluate several feasible values for the base-stock levels (S_1, S_2). Assume that S_i must be an integer.

a) Explain why $S_1^* + S_2^* \leq 2$.

b) For each possible solution (S_1, S_2) below, write the expected values of the state variables Q_i, Y_{ij}, IL_i^+, and IL_i^-, and then write the expected cost $g(S_1, S_2)$.

1. $(S_1, S_2) = (0, 0)$
2. $(S_1, S_2) = (1, 1)$
3. $(S_1, S_2) = (1, 0)$
4. $(S_1, S_2) = (2, 0)$

(The cases in which $(S_1, S_2) = (0, 1)$ or $(0, 2)$ are similar to the cases above, so we'll skip them.)

Hint 1: If $S_i = 0$, that does *not* mean that stage i never orders!

Hint 2: To check your cost functions, we'll tell you the following: If $c_i = h_i = p_i = 1$, $c_{ij} = 3$, and $q_i = 0.5$ for all $i = 1, 2$, then $g(0, 0) = 2$, $g(1, 1) = 2$, $g(1, 0) = 2.25$, and $g(2, 0) = 3.5$. Note, however, that these parameters do not satisfy the assumptions on page 238.

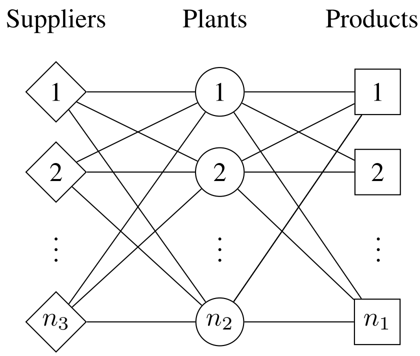


Figure 7.7 Three-stage flexibility structure for Problem 7.10.

c) Find an instance for which $(S_1^*, S_2^*) = (1, 1)$. Your instance must satisfy the assumptions on page 238.

d) Find a *symmetric* instance for which $(S_1^*, S_2^*) = (1, 0)$. Your instance must satisfy the assumptions on page 238. A symmetric instance is one for which the parameters for the two retailers are identical ($c_1 = c_2$, $h_1 = h_2$, etc.). (It's a little surprising that a symmetric instance can produce a nonsymmetric solution, but it can.)

e) Prove or disprove the following claim: $g(2, 0) \geq g(1, 1)$ for all instances that satisfy the assumptions on page 238.

7.10 (Three-Stage Flexibility) Consider the three-stage supply chain flexibility design problem pictured in Figure 7.7.

There are n_1 products, n_2 plants, and n_3 suppliers. In the full-flexibility structure, each product can be produced at any plant using raw materials sourced from any supplier. We assume that each unit of product consumes one unit of material from each supplier and uses one unit of capacity at each plant. We assume further that the production capacities at the plants are C_j, $j = 1, \ldots, n_2$, and that the suppliers have a limited amount B_k of raw materials, $k = 1, \ldots, n_3$. The demand for each product is random and is denoted by the random variable D_i, $i = 1, \ldots, n_1$.

a) Derive an expression for the expected sales in the full-flexibility structure.

b) Let y_{ijk} be a decision variable representing the amount of raw materials from supplier k used to produce product i at plant j. Formulate the flexibility design problem for this three-stage supply chain.

7.11 (Capacity Investment) Recall the formulation of the flexibility design problem (7.32)–(7.37). Suppose now that the capacity is also a decision, to be made jointly with the network design problem. In particular, the capacities C_j are first-stage decision variables, together with the flexibility investment variables x_{ij}. We assume a linear investment cost function for the capacity, with constant marginal investment cost v_j per unit.

a) Write the new objective function after adding the capacity-investment cost term.

b) Discuss a method for solving this new problem.

7.12 (Auto Repair) A small car repair shop has four certified technicians, Irene, Larry, Max, and Suzanne. The shop specializes in four types of vehicles, labeled A, B, C, and

D. Each technician has been trained to repair one type of vehicle. On average, the repair of each type of vehicle takes 4 hours. It is estimated that the number of customers who want a type-A vehicle fixed during a given week is equally likely to be 8, 10, or 12. That is, the probability is $\frac{1}{3}$ for each of the possible outcomes, 8, 10, or 12 customers. Each of the other three vehicle types has the same demand distribution for repairs. Furthermore, the demands are iid across time and type. Each technician has a nominal work week of 40 hours at \$55 per hour and will be paid for 40 hours even if he or she works fewer than 40 hours in the week. But if a technician works more than 40 hours, then the overtime rate is 150% of the normal pay. The overtime rate applies only to the hours in excess of 40.

a) Calculate the expected yearly cost of the dedicated system (in which Irene, Larry, Max, and Suzanne can only service vehicles of type A, B, C, and D, respectively).

b) After reading this chapter, the repair shop's manager decides to try a flexible system. Suppose first that the manager uses a chaining structure in which Irene will also be trained to repair type-B vehicles, Larry to repair type-C vehicles, Max to repair type-D vehicles, and Suzanne to repair type-A vehicles. Calculate the expected yearly cost of overtime for this system.

Note: Given the assumptions made in this problem, for each realization of the random demands, it is possible to calculate the total overtime cost based on the total hours required without determining the assignments of vehicles to technicians.

c) Suppose instead that the manager chooses a full-flexibility system in which every technician is trained to repair all four vehicle models. Calculate the expected yearly cost of overtime for this system.

d) Determine the optimal assignment of vehicles to technicians under both the chaining and full flexibility structures if the demands are 12, 8, 10, and 12 for vehicles of types A–D, respectively.

e) Suppose the cost of training one technician to repair one new vehicle type is \$10,000. What is the expected number of years until the shop recoups the investment cost to convert the dedicated system to the partial flexibility system with the chaining structure? What about the full flexibility system?

7.13 (Auto Repair, Part 2) Consider again the auto repair shop in Problem 7.12. Suppose that the number of repairs of type-A vehicles in a given week has a normal distribution with a mean of 22 and a standard deviation of 3, while the number of repairs of vehicles of types B–D is deterministic and equal to 6 for each type. Calculate the expected yearly cost of overtime for both the chaining structure (as described in Problem 7.12(b)) and the full flexibility system.

Note: Unlike in Problem 7.12, in this problem you must determine the optimal assignment of vehicles to technicians for each realization of the demand in order to calculate the expected overtime cost for the chaining system.

7.14 (Max-Flow Formulation) The production-allocation problem (7.17)–(7.21) can also be formulated using a max-flow formulation.

a) Formulate the problem as a max-flow problem, using the notation already defined in the chapter.

b) Research has shown that when demands are independent, chaining can achieve most (roughly 97%) of the benefits of full flexibility as the number of nodes n

approaches ∞. Show that if the demands are correlated, the situation can be very different.

7.15 (Lemma 7.7 Only Holds in Expectation) Develop a counterexample to show that, although Lemma 7.7 holds in expectation, the result does not hold for every individual demand instance.

CHAPTER 8

FACILITY LOCATION MODELS

8.1 INTRODUCTION

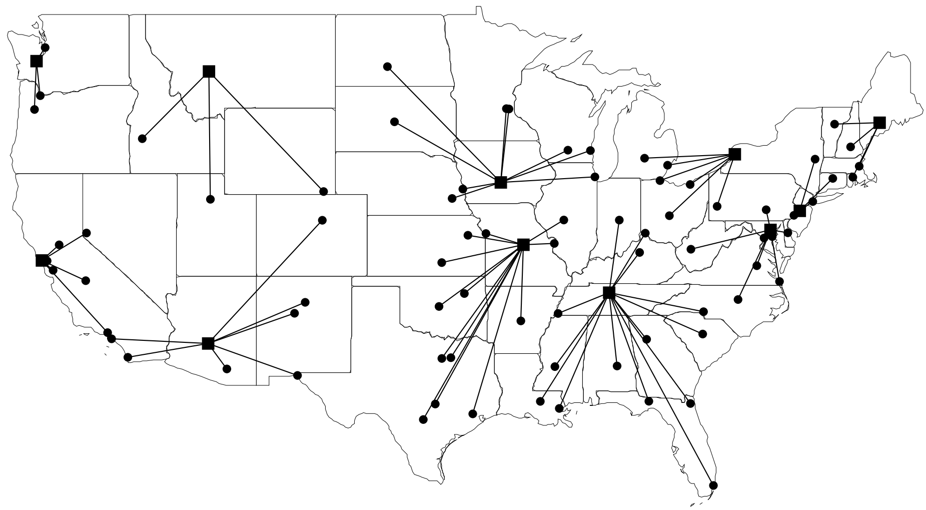
One of the major strategic decisions faced by firms is the number and locations of factories, warehouses, retailers, or other physical facilities. This is the purview of a large class of models known as *facility location problems*. The key trade-off in most facility location problems is between the facility cost and customer service. If we open a lot of facilities (Figure 8.1(a)), we incur high facility costs (to build and maintain them), but we can provide good service since most customers are close to a facility. On the other hand, if we open few facilities (Figure 8.1(b)), we reduce our facility costs but must travel farther to reach our customers (or they to reach us).

Most (but not all) location problems make two related sets of decisions: (1) where to locate, and (2) which customers are assigned or allocated to which facilities. Therefore, facility location problems are also sometimes known as *location–allocation problems*.

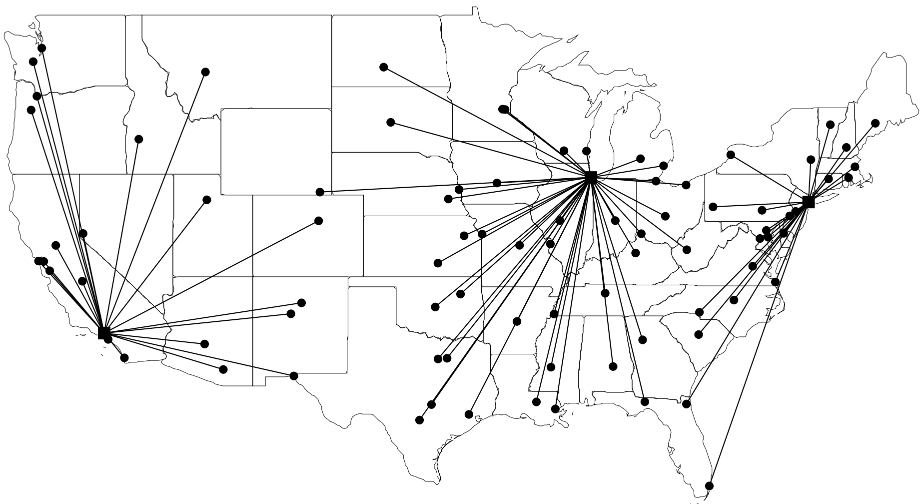
A huge range of approaches has been considered for modeling facility location decisions. These differ in terms of how they model facility costs (for example, some include the costs explicitly, while others impose a constraint on the number of facilities to be opened) and how they model customer service (for example, some include a transportation cost, while others require all or most facilities to be *covered*—that is, served by a facility that is within some specified distance). Facility location problems come in a great variety of flavors based on what types of facilities are to be located, whether the facilities are capacitated, which (if any) elements of the problem are stochastic, what topology the facilities may be located

Fundamentals of Supply Chain Theory, Second Edition. Lawrence V. Snyder and Zuo-Jun Max Shen.

Companion website: www.wiley.com/go/Snyder/SupplyChainTheory



(a) Many facilities open.



(b) Few facilities open.

Figure 8.1 Facility location configurations. Squares represent facilities; circles represent customers.

on (e.g., on the plane, in a network, or at discrete points), how distances or transportation costs are measured, and so on. Several excellent textbooks provide additional material for the interested reader; for example, see Mirchandani and Francis (1990), Drezner (1995a), Drezner and Hamacher (2002), or Daskin (2013). For an annotated bibliography of papers on facility location problems, see ReVelle et al. (2008b). The book by Eiselt and Marianov (2011) contains chapters on a number of seminal papers in facility location, each describing the original contribution as well as later extensions.

In addition to supply chain facilities such as plants and warehouses, location models have been applied to public sector facilities such as bus depots and fire stations, as well as to telecommunications hubs, satellite orbits, bank accounts, and other items that are not really "facilities" at all. In addition, many operations research problems can be formulated as facility location problems or have subproblems that resemble them. Facility location problems are often easy to state and formulate but are difficult to solve; this makes them a popular testing ground for new optimization tools. For all of these reasons, facility location problems are an important topic in operations research, and in supply chain management in particular, in both theoretical and applied work.

In this chapter, we will begin by discussing a classical facility location model, the uncapacitated fixed-charge location problem (UFLP), in Section 8.2. The UFLP and its descendants have been deployed more widely in supply chain management than perhaps any other location model. One reason for this is that the UFLP is very flexible and, although it is NP-hard, lends itself to a variety of effective solution methods. Another reason is that the UFLP includes explicit costs for both key elements of the problem—facilities and customer service—and is therefore well suited to supply chain applications.

In Section 8.3, we discuss other so-called *minisum* models (in particular, the p-median problem and a capacitated version of the UFLP), and in Section 8.4, we discuss *covering* models (including the p-center, set covering, and maximal covering problems). We briefly discuss a variety of other deterministic facility location problems in Section 8.5. In Section 8.6, we introduce stochastic and robust models for facility location under uncertainty. We then discuss models for *network design*—a close cousin of facility location—in Section 8.7.

8.2 THE UNCAPACITATED FIXED-CHARGE LOCATION PROBLEM

8.2.1 Problem Statement

The *uncapacitated fixed-charge location problem* (UFLP) chooses facility locations in order to minimize the total cost of building the facilities and transporting goods from facilities to customers. The UFLP makes location decisions for a single echelon, and the facilities in that echelon are assumed to serve facilities in a downstream echelon, all of whose locations are fixed. We will tend to refer to the facilities in the upstream echelon as *distribution centers* (DCs) or *warehouses* and to those in the downstream echelons as *customers*. However, the model is generic, and the two echelons may instead contain other types of facilities—for example, factories and warehouses, or regional and local DCs, or even fire stations and homes. Sometimes it's also useful to think of an upstream echelon, again with fixed location(s), that serves the DCs.

Each potential DC location has a *fixed cost* that represents building (or leasing) the facility; the fixed cost is independent of the volume that passes through the DC. There is a *transportation cost* per unit of product shipped from a DC to each customer. There is a single product. The DCs have no capacity restrictions—any amount of product can be handled by any DC. (We'll relax this assumption in Section 8.3.1.) The problem is to choose facility locations to minimize the fixed cost of building facilities plus the transportation cost to transport product from DCs to customers, subject to constraints requiring every customer to be served by some open DC.

As noted above, the key trade-off in the UFLP is between fixed and transportation costs. If too few facilities are open, the fixed cost is small, but the transportation cost is large because many customers will be far from their assigned facility. On the other hand, if too many facilities are open, the fixed cost is large, but the transportation cost is small. The UFLP tries to find the right balance, and to optimize not only the number of facilities, but also their locations.

8.2.2 Formulation

Define the following notation:

Sets

I = set of customers

J = set of potential facility locations

Parameters

h_i = annual demand of customer $i \in I$

c_{ij} = cost to transport one unit of demand from facility $j \in J$ to customer $i \in I$

f_j = fixed annual cost to open a facility at site $j \in J$

Decision Variables

x_j = 1 if facility j is opened, 0 otherwise

y_{ij} = the fraction of customer i's demand that is served by facility j

The transportation costs c_{ij} might be of the form $k \times$ distance for some constant k (if the shipping company charges k per mile per unit) or may be more arbitrary (for example, based on airline ticket prices, which are not linearly related to distance). In the former case, distances may be computed in a number of ways:

- *Euclidean distance*: The distance between (a_1, b_1) and (a_2, b_2) is given by

  $$\sqrt{(a_1 - a_2)^2 + (b_1 - b_2)^2}.$$

  The Euclidean distance metric is also known as the ℓ_2 *norm*. This is an intuitive measure of distance but is not usually applicable in supply chain contexts because Cartesian coordinates are not useful for describing real-world locations.

- *Manhattan or rectilinear metric*: The distance is given by

  $$|a_1 - a_2| + |b_1 - b_2|.$$

  This metric assumes that travel is only possible parallel to the x- or y-axis, e.g., travel along city streets. It is also known as the ℓ_1 *norm*.

- *Great circle*: This method for calculating distances takes into account the curvature of the earth and, more importantly, takes latitudes and longitudes as inputs and returns distances in miles or kilometers. Great circle distances assume that travel occurs over a great circle, the shortest route over the surface of a sphere. Let (α_1, β_1) and (α_2, β_2) be the latitude and longitude of two points in radians, and let $\Delta\alpha \equiv \alpha_1 - \alpha_2$ and $\Delta\beta \equiv \beta_1 - \beta_2$ be the differences in the latitude and longitude (respectively). Then the great circle distance between the two points is given by

$$2r \arcsin\left(\sqrt{\sin^2\left(\frac{\Delta\alpha}{2}\right) + \cos\alpha_1 \cos\alpha_2 \sin^2\left(\frac{\Delta\beta}{2}\right)}\right), \tag{8.1}$$

where r is the radius of the Earth, approximately 3958.76 miles or 6371.01 km (on average), and the trigonometric functions are assumed to use radians.

A simpler formula, known as the *spherical law of cosines*, sets the distance equal to

$$r \arccos\left(\sin\alpha_1 \sin\alpha_2 + \cos\alpha_1 \cos\alpha_2 \cos\left(\Delta\beta\right)\right) \tag{8.2}$$

and is nearly as accurate as (8.1) except when the distance between the two points is very small. (See Problem 8.44.)

- *Highway/network*: The distance is computed as the shortest path within a network, for example, the US highway network. This is usually the most accurate method for calculating distances in a supply chain context. However, since they require data on the entire road network, they must be obtained from geographic information systems (GIS) or from online services such as Mapquest or Google Maps. (In contrast, the distance measures above can be calculated from simple formulas using only the coordinates of the facilities and customers.)
- *Matrix*: Sometimes a matrix containing the distance between every pair of points is given explicitly. This is the most general measure, since all others can be considered a special case. It is also the only possible measure when the cost structure exhibits no particular pattern—for example, when they are based on airline ticket prices.

In general, we won't be concerned with how transportation costs are computed—we'll assume they are given to us already as the parameters c_{ij}.

The UFLP is formulated as follows:

$$\text{(UFLP)} \quad \text{minimize} \quad \sum_{j\in J} f_j x_j + \sum_{i\in I}\sum_{j\in J} h_i c_{ij} y_{ij} \tag{8.3}$$

$$\text{subject to} \quad \sum_{j\in J} y_{ij} = 1 \qquad \forall i \in I \tag{8.4}$$

$$y_{ij} \le x_j \qquad \forall i \in I, \forall j \in J \tag{8.5}$$

$$x_j \in \{0,1\} \qquad \forall j \in J \tag{8.6}$$

$$y_{ij} \ge 0 \qquad \forall i \in I, \forall j \in J \tag{8.7}$$

Formulations very similar to this were originally proposed by Manne (1964) and Balinski (1965). The objective function (8.3) computes the total (fixed plus transportation) cost. In the discussion that follows, we'll use z^* to denote the optimal objective value of (UFLP). Constraints (8.4) require the full amount of every customer's demand to be assigned, to one or more facilities. These are often called *assignment constraints*. Constraints (8.5) prohibit a customer from being assigned to a facility that has not been opened. These are often called *linking constraints*. Constraints (8.6) require the location (x) variables to be binary, and constraints (8.7) require the assignment (y) variables to be nonnegative.

Constraints (8.4) and (8.7) together ensure that $0 \le y_{ij} \le 1$. In fact, it is always optimal to assign each customer solely to its nearest open facility. (Why?) Therefore, there always exists an optimal solution in which $y_{ij} \in \{0,1\}$ for all $i \in I,\ j \in J$. It is therefore appropriate to think of the y_{ij} as binary variables and to talk about "the facility to which customer i is assigned."

Another way to write constraints (8.5) is

$$\sum_{i \in I} y_{ij} \le |I| x_j \qquad \forall j \in J. \tag{8.8}$$

If $x_j = 1$, then y_{ij} can be 1 for any or all $i \in I$, while if $x_j = 0$, then y_{ij} must be 0 for all i. These constraints are equivalent to (8.5) for the IP. But the LP relaxation is weaker (i.e., it provides a weaker bound) if constraints (8.8) are used instead of (8.5). This is because there are solutions that are feasible for the LP relaxation with (8.8) that are not feasible for the LP relaxation with (8.5). To take a trivial example, suppose there are 2 facilities and 10 customers with equal demand, and suppose each facility serves 5 customers in a given solution. Then it is feasible to set $x_1 = x_2 = \frac{1}{2}$ for the problem with (8.8) but not with (8.5). Since the feasible region for the problem with (8.8) is larger than that for the problem with (8.5), its objective value is no greater. It is important to understand that the IPs have the *same* optimal objective value, but the LPs have different values—one provides a weaker LP bound than the other.

The UFLP is NP-hard (Garey and Johnson 1979). A large number of solution methods have been proposed in the literature over the past several decades, both exact algorithms and heuristics. Some of the earliest exact algorithms involve simply solving the IP using branch-and-bound. Today, this would mean solving (UFLP) as-is using CPLEX, Gurobi, or another off-the-shelf IP solver, although such general-purpose solvers did not exist when the UFLP was first formulated. This approach works quite well using modern solvers, in part because the LP relaxation of (UFLP) is usually extremely tight, and in fact it often results in all-integer solutions "for free" (Morris 1978). (ReVelle and Swain (1970) discuss this property in the context of a related problem, the p-median problem.) Current versions of CPLEX or Gurobi can solve instances of the UFLP with thousands of potential facility sites in a matter of minutes. However, when it was first proposed that branch-and-bound be used to solve the UFLP (by Efroymson and Ray (1966)), IP technology was much less advanced, and this approach could only be used to solve problems of modest size. Therefore, a number of other optimal approaches were developed. Two of these—Lagrangian relaxation and a dual-ascent method called DUALOC—are discussed in Sections 8.2.3 and 8.2.4, respectively. Many other IP techniques, such as Dantzig–Wolfe or Benders decomposition, have also been successfully applied to the UFLP (e.g., Balinski (1965) and Swain (1974)). We discuss heuristic methods for the UFLP in Section 8.2.5.

8.2.3 Lagrangian Relaxation

8.2.3.1 Introduction One of the methods that has proven to be most effective for the UFLP and other location problems is Lagrangian relaxation, a standard technique for integer programming (as well as other types of optimization problems). The basic idea behind Lagrangian relaxation is to remove a set of constraints to create a problem that's easier to solve than the original. But instead of just removing the constraints, we include them in the objective function by adding a term that penalizes solutions for violating the constraints. This process gives a *lower bound* on the optimal objective value of the UFLP, but it does not necessarily give a feasible solution. Feasible solutions must be found using some other method (to be described below); each feasible solution provides an *upper bound* on the optimal objective value. When the upper and lower bounds are close (say, within 1%), we know that the feasible solution we have found is close to optimal.

For more details on Lagrangian relaxation, see Appendix D.1. See also Fisher (1981, 1985) for excellent overviews. Lagrangian relaxation was proposed as a method for solving a UFLP-like problem by Cornuejols et al. (1977).

We want to use Lagrangian relaxation on the UFLP formulation given in Section 8.2.2. The question is, which constraints should we relax? There are only two options: (8.4) and (8.5). (Constraints (8.6) and (8.7) can't be relaxed using Lagrangian relaxation.) Relaxing either (8.4) or (8.5) results in a problem that is quite easy to solve, and both relaxations produce the same bound (for reasons discussed below). But relaxing (8.4) involves relaxing fewer constraints, which is generally preferable (also for reasons that will be discussed below). Therefore, we will relax constraints (8.4), although in Section 8.2.3.8 we will briefly discuss what happens when constraints (8.5) are relaxed.

8.2.3.2 *Relaxation* We relax constraints (8.4), removing them from the problem and adding a penalty term to the objective function:

$$\sum_{i\in I} \lambda_i \left(1 - \sum_{j\in J} y_{ij}\right)$$

The λ_i are called *Lagrange multipliers*. There is one for each relaxed constraint. Their purpose is to ensure that violations in the constraints are penalized by just the right amount—more on this later. We'll use λ to represent the vector of λ_i values.

For now, assume λ is fixed. Relaxing constraints (8.4) gives us the following problem, known as the *Lagrangian subproblem*:

$$\begin{aligned}
(\text{UFLP-LR}_\lambda) \quad \text{minimize} \quad & \sum_{j\in J} f_j x_j + \sum_{i\in I}\sum_{j\in J} h_i c_{ij} y_{ij} + \sum_{i\in I} \lambda_i \left(1 - \sum_{j\in J} y_{ij}\right) \\
= & \sum_{j\in J} f_j x_j + \sum_{i\in I}\sum_{j\in J} (h_i c_{ij} - \lambda_i) y_{ij} + \sum_{i\in I} \lambda_i \qquad (8.9)
\end{aligned}$$

$$\begin{aligned}
\text{subject to} \quad & y_{ij} \le x_j && \forall i \in I, \forall j \in J && (8.10)\\
& x_j \in \{0,1\} && \forall j \in J && (8.11)\\
& y_{ij} \ge 0 && \forall i \in I, \forall j \in J && (8.12)
\end{aligned}$$

(The subscript λ on the problem name reminds us that this problem depends on λ as a parameter.) Since the λ_i are all constants, the last term of (8.9) can be ignored during the optimization.

How can we solve this problem? It turns out that the problem is quite easy to solve by inspection—we don't need to use an IP solver or any sort of complicated algorithm.

Suppose that we set $x_j = 1$ for a given facility j. By constraints (8.10), setting $x_j = 1$ allows y_{ij} to be set to 1 for any $i \in I$. For which i would y_{ij} be set to 1 in an optimal solution to the problem? Since this is a minimization problem, y_{ij} would be set to 1 for all i such that $h_i c_{ij} - \lambda_i < 0$. So if x_j were set to 1, the *benefit* (or contribution to the objective function) would be

$$\beta_j = \sum_{i \in I} \min\{0, h_i c_{ij} - \lambda_i\}. \tag{8.13}$$

Now the question is, which x_j should be set to 1? It's optimal to set $x_j = 1$ if and only if $\beta_j + f_j < 0$; that is, if the benefit of opening the facility outweighs its fixed cost. Theorem 8.1 summarizes these conclusions.

Theorem 8.1 *Let*

$$\bar{x}_j = \begin{cases} 1, & \textit{if } \beta_j + f_j < 0 \\ 0, & \textit{otherwise} \end{cases} \tag{8.14}$$

$$\bar{y}_{ij} = \begin{cases} 1, & \textit{if } \bar{x}_j = 1 \textit{ and } h_i c_{ij} - \lambda_i < 0 \\ 0, & \textit{otherwise.} \end{cases} \tag{8.15}$$

Then $(\bar{x}, \bar{y})$ *is an optimal solution for* (UFLP-LR$_\lambda$)*, and it has an objective value of*

$$z_{\text{LR}}(\lambda) = \sum_{j \in J} \min\{0, \beta_j + f_j\} + \sum_{i \in I} \lambda_i.$$

Notice that in optimal solutions to (UFLP-LR$_\lambda$), customers may be assigned to 0 or more than 1 facility since the constraints requiring exactly one facility per customer have been relaxed.

Why is this problem so much easier to solve than the original problem? The answer is that (UFLP-LR$_\lambda$) decomposes by j, in the sense that we can focus on each $j \in J$ individually since there are no constraints tying them together. In the original problem, constraints (8.4) tied the js together—we could not make a decision about y_{ij} without also making a decision about y_{ik} since i had to be assigned to exactly one facility.

The method for solving (UFLP-LR$_\lambda$) is summarized in Algorithm 8.1.

8.2.3.3 Lower Bound We've now solved (UFLP-LR$_\lambda$) for given λ_i. How does this help us? Well, from Theorem D.1, we know that, for any λ, the optimal objective value of (UFLP-LR$_\lambda$) is a lower bound on the optimal objective value for the original problem:

$$z_{\text{LR}}(\lambda) \le z^*. \tag{8.16}$$

The point of Lagrangian relaxation is not to generate feasible solutions, since the solutions to (UFLP-LR$_\lambda$) will generally be infeasible for (UFLP). Instead, the point is to generate good (i.e., high) lower bounds in order to prove that a feasible solution we've found some other way is good. For example, if we've found a feasible solution for the UFLP (using any method at all) whose objective value is 1005 and we've also found a λ so that $z_{\text{LR}}(\lambda) = 1000$, then we know our solution is no more than $(1005 - 1000)/1000$ = 0.5% away from optimal. (It may in fact be *exactly* optimal, but given these two bounds, we can only say it's within 0.5%.)

Algorithm 8.1 Solve (UFLP-LR$_\lambda$)

input Lagrange multipliers λ
for all $j \in J$ **do** ▷ Main loop
  $\beta_j \leftarrow \sum_{i\in I} \min\{0, h_i c_{ij} - \lambda_i\}$ ▷ Calculate benefit
  if $\beta_j + f_j < 0$ **then** ▷ Check benefit vs. fixed cost
    $\bar{x}_j \leftarrow 1$ ▷ Open j
    for all $i \in I$ **do**
      if $h_i c_{ij} - \lambda_i < 0$ **then** $\bar{y}_{ij} \leftarrow 1$ **else** $\bar{y}_{ij} \leftarrow 0$ **end if**
    end for
  else
    $\bar{x}_j \leftarrow 0$ ▷ Do not open j
    for all $i \in I$ **do**
      $\bar{y}_{ij} \leftarrow 0$
    end for
  end if
end for
$z_{\text{LR}}(\lambda) \leftarrow \sum_{j\in J} \min\{0, \beta_j + f_j\} + \sum_{i\in I} \lambda_i$ ▷ Calculate objective function
return $\bar{x}$, $\bar{y}$, $z_{\text{LR}}(\lambda)$

Now, if we pick λ at random, we're not likely to get a particularly good bound—that is, $z_{\text{LR}}(\lambda)$ won't be close to z^*. We have to choose λ cleverly so that we get the best possible bound—so that $z_{\text{LR}}(\lambda)$ is as large as possible. That is, we want to solve problem (LR) given in (D.8), which, for the UFLP, can be written as follows:

$$\text{(LR)} \qquad \max_{\lambda} \left\{ \begin{array}{ll} \min_{x,y} & \sum_{j\in J} f_j x_j + \sum_{i\in I} (h_i c_{ij} - \lambda_i) y_{ij} + \sum_{i\in I} \lambda_i \\ \text{s.t.} & y_{ij} \le x_j \quad \forall i \in I, \forall j \in J \\ & x_j \in \{0,1\} \quad \forall j \in J \\ & y_{ij} \ge 0 \quad \forall i \in I, \forall j \in J \end{array} \right\} \tag{8.17}$$

We'll talk more later about how to solve this problem. For now, let's assume we know the optimal λ^* and that the optimal objective value is $z_{\text{LR}} \equiv z_{\text{LR}}(\lambda^*)$. How large can z_{LR} be? Theorem D.1 tells us it cannot be larger than z^*, but how close can it get? The answer turns out to be related to the LP relaxation of the problem. From Theorem D.2, we have

$$z_{\text{LP}} \le z_{\text{LR}}, \tag{8.18}$$

where z_{LP} is the optimal objective value of the LP relaxation of (UFLP) and z_{LR} is the optimal objective value of (LR).

Combining (8.16) and (8.18), we now know that

$$z_{\text{LP}} \le z_{\text{LR}} \le z^*. \tag{8.19}$$

For most problems, $z_{\text{LP}} \lneqq z^*$, so where in the gap does z_{LR} fall? An IP is said to have the *integrality property* if its LP relaxation naturally has an all-integer optimal solution. You should be able to convince yourself that (UFLP-LR$_\lambda$) has the integrality property for all λ since it is never better to set x and y to fractional values. Therefore, the following is a corollary to Lemma D.3:

Corollary 8.2 *For the UFLP,* $z_{\text{LP}} = z_{\text{LR}}$.

Combining (8.19) and Corollary 8.2, we have

$$z_{\text{LR}} = z_{\text{LP}} \le z^*.$$

This means that if the LP relaxation bound from the UFLP is not very tight, the Lagrangian relaxation bound won't be very tight either. Fortunately, as noted in Section 8.2.2, the UFLP tends to have very tight LP relaxation bounds. This raises the question of why we'd want to use Lagrangian relaxation at all since the LP bound is just as tight.

There are several possible answers to this question. The first is that when Lagrangian relaxation was first applied to the UFLP, computer implementations of the simplex method were quite inefficient, and even the LP relaxation of the UFLP could take a long time to solve, whereas the Lagrangian subproblem could be solved quite quickly. Recent implementations of the simplex method, however (for example, recent versions of CPLEX), are much more efficient and are able to solve reasonably large instances of the UFLP—LP or IP—pretty quickly. Nevertheless, Lagrangian relaxation is still an important tool for solving the UFLP. One advantage of this method is that it can often be modified to solve extensions of the UFLP that IP solvers can't solve—for example, nonlinear, nonconvex problems like the location model with risk pooling (LMRP), which we discuss in Section 12.2.

It is important to distinguish between z_{LR} (the best possible lower bound achievable by Lagrangian relaxation) and $z_{\text{LR}}(\lambda)$ (the lower bound achieved at a given iteration of the procedure). At any given iteration, we have

$$z_{\text{LR}}(\lambda) \le z_{\text{LR}} = z_{\text{LP}} \le z^* \le z(x, y), \tag{8.20}$$

where $z_{\text{LR}}(\lambda)$ is the objective value of the Lagrangian subproblem for the particular λ at the current iteration, and $z(x, y)$ is the objective value of the particular feasible solution (x, y) found at the current iteration.

8.2.3.4 Upper Bound Now that we've obtained a lower bound on the optimal objective of (UFLP) using (UFLP-LR$_\lambda$), we need to find an upper bound. Upper bounds come from feasible solutions to (UFLP). How can we build good feasible solutions? One way would be using construction and/or improvement heuristics like those described in Section 8.2.5. But we'd like to take advantage of the information contained in the solutions to (UFLP-LR$_\lambda$); that is, we'd like to convert a solution to (UFLP-LR$_\lambda$) into one for (UFLP). Remember that solutions to (UFLP-LR$_\lambda$) consist of a set of facility locations (identified by the x variables) and a set of assignments (identified by the y variables). It is the y variables that make the solution infeasible for (UFLP), since customers might be assigned to 0 or more than 1 facility. (If every customer happens to be assigned to exactly 1 facility, the solution is also feasible for (UFLP). In fact, it is *optimal* for (UFLP) since it has the same objective value for both (UFLP-LR$_\lambda$), which provides a lower bound, and (UFLP), which provides an upper bound. But we can't expect this to happen in general.)

Generating a feasible solution for (UFLP) is easy: We just open the facilities that are open in the solution to (UFLP-LR$_\lambda$) and then assign each customer to its nearest open facility. (See Algorithm 8.2.) The resulting solution is feasible and provides an upper bound on the optimal objective value of (UFLP). Sometimes an improvement heuristic (like the swap or neighborhood search heuristics discussed in Section 8.3.2.3) is applied to each feasible solution found, but this is optional.

Algorithm 8.2 Get feasible solution for UFLP from solution to (UFLP-LR$_\lambda$)

input location vector $\bar{x}$ for (UFLP-LR$_\lambda$)
$x \leftarrow \bar{x}$ ▷ Open facilities in lower bound solution
for all $i \in I$ **do** ▷ Main loop
    $j^* \leftarrow \text{argmin}_{\bar{x}_j=1}\{c_{ij}\}$ ▷ Find nearest open facility to i
    $y_{ij^*} \leftarrow 1$ ▷ Assign i to j^*
    for all $j \in J, j \neq j^*$ **do**
        $y_{ij} \leftarrow 0$
    end for
end for
$z(x,y) \leftarrow \sum_{j\in J} f_j x_j + \sum_{i\in I}\sum_{j\in J} h_i c_{ij} y_{ij}$ ▷ Calculate obj. function
return x, y, $z(x,y)$

8.2.3.5 Updating the Multipliers Each λ gives a single lower bound and (using the method in Section 8.2.3.4) a single upper bound. The Lagrangian relaxation process involves many iterations, each using a different value of λ, in the hopes of tightening the bounds. It would be impractical to try every possible value of λ; we want to choose λ cleverly.

Using the logic of Section D.1.3, if λ_i is too small, there's no real incentive to set the y_{ij} variables to 1 since the penalty will be small. On the other hand, if λ_i is too large, there will be an incentive to set *lots* of y_{ij} variables to 1, making the term inside the parentheses negative and the overall penalty large and negative. (Remember that (UFLP-LR$_\lambda$) is a minimization problem.) By changing λ_i, we'll encourage fewer or more y_{ij} variables to be 1.

So:

- If $\sum_{j\in J} y_{ij} = 0$, then λ_i is too small; it should be increased.
- If $\sum_{j\in J} y_{ij} > 1$, then λ_i is too large; it should be decreased.
- If $\sum_{j\in J} y_{ij} = 1$, then λ_i is just right; it should not be changed.

Here's another way to see the effect of changing λ_i. Remember that if $x_j = 1$ in the solution to (UFLP-LR$_\lambda$), y_{ij} will be set to 1 if

$$h_i c_{ij} - \lambda_i < 0.$$

Increasing λ_i makes this hold for more facilities j, while decreasing it makes it hold for fewer.

There are several ways to make these adjustments to λ. Perhaps the most common is *subgradient optimization*, discussed in Section D.1.3. For the UFLP, the step size at iteration t (denoted Δ^t) is given by

$$\Delta^t = \frac{\alpha^t(\text{UB} - z_{\text{LR}}(\lambda^t))}{\sum_{i\in I}\left(1 - \sum_{j\in J} y_{ij}\right)^2}, \tag{8.21}$$

where $z_{\text{LR}}(\lambda^t)$ is the lower bound found at iteration t, UB is the best upper bound found (i.e., the objective value of the best feasible solution found so far), and α^t is a constant that is generally set to 2 at iteration 1 and divided by 2 after a given number (say 20) of consecutive iterations have passed during which the best known lower bound has not improved. The step direction for iteration i is simply given by

$$1 - \sum_{j \in J} y_{ij}$$

(the violation in the constraint).

To obtain the new multipliers (call them λ^{t+1}) from the old ones (λ^t), we set

$$\lambda_i^{t+1} = \lambda_i^t + \Delta^t \left(1 - \sum_{j \in J} y_{ij}\right). \tag{8.22}$$

Note that since $\Delta^t > 0$, this follows the rules given above: If $\sum_{j \in J} y_{ij} = 0$, then λ_i increases; if $\sum_{j \in J} y_{ij} > 1$, then λ_i decreases; and if $\sum_{j \in J} y_{ij} = 1$, then λ_i stays the same.

The process of solving (UFLP-LR$_\lambda$), finding a feasible solution, and updating λ is continued until some of criteria are met. (See Section D.1.4.)

At the first iteration, λ can be initialized using a variety of ways: For example, set $\lambda_i = 0$ for all i, set it to some random number, or set it according to some other ad-hoc rule.

If the Lagrangian procedure stops before the upper and lower bounds are sufficiently close to each other, we can use branch-and-bound to close the optimality gap; see Section D.1.6. The Lagrangian procedure is summarized in Section D.1.7.

8.2.3.6 Summary The Lagrangian relaxation method for the UFLP is summarized in the pseudocode in Algorithm 8.3. In the pseudocode, $(\bar{x}, \bar{y})$ represents an optimal solution to (UFLP-LR$_\lambda$), (x, y) represents a feasible solution to (UFLP), and $(x^{\text{UB}}, y^{\text{UB}})$ represents the current best solution for (UFLP). Note that in step 29, other termination criteria can be used, instead or in addition.

☐ **EXAMPLE 8.1**

The instance pictured in Figure 8.1 is the 88-node instance from Daskin (1995). It consists of the capitals of the 48 continental United States, plus Washington, DC, plus the 50 largest cities in the 1990 US census, minus duplicates. In this instance, $I = J$: Every node is both a customer and a potential facility site. Demands h_i are set equal to the city populations divided by 1000; fixed costs f_j are set equal to the median home value; and transportation costs c_{ij} are set equal to 0.5 times the great circle distance between i and j. (The full data set, along with other related data sets, are available on the book's companion web site.)

The optimal solution locates five facilities, in Houston, TX; Philadelphia, PA; Detroit, MI; Fresno, CA; and Topeka, KS. The total cost of this solution is \$783,813, with fixed and transportation costs of \$521,713 and \$262,100, respectively. We obtained this solution using the Lagrangian relaxation algorithm discussed in this section, implemented in MATLAB, with a total CPU time of less than 2 seconds on a laptop computer.

In case you're curious: The 9-facility solution shown in Figure 8.1(a) has a total cost of \$1,480,059 (\$954,600 fixed cost plus \$525,459 transportation cost), while the

Algorithm 8.3 Lagrangian relaxation algorithm for UFLP

input initial multipliers λ^1, initial constant α^0, α-halving constant γ, optimality tolerance κ, iteration limit ζ
$t \leftarrow 1$, LB $\leftarrow -\infty$, UB $\leftarrow \infty$, `NonImprCtr` $\leftarrow 0$ ▷ Initialization
repeat ▷ Main loop
  solve (UFLP-LR$_\lambda$) using Algorithm 8.1 with input λ^t ▷ Lower bound
  $(\bar{x}, \bar{y})$, $z_{\text{LR}}(\lambda^t) \leftarrow$ output of Algorithm 8.1
  if $z_{\text{LR}}(\lambda^t) >$ LB **then** ▷ Compare to best-known lower bound
    LB $\leftarrow z_{\text{LR}}(\lambda^t)$
    `NonImprCtr` $\leftarrow 0$ ▷ Reset non-improvement counter
  else
    `NonImprCtr` $\leftarrow$ `NonImprCtr` $+ 1$ ▷ Increment non-impr. counter
    if `NonImprCtr` $= \gamma$ **then** ▷ Check whether to halve α
      $\alpha^t \leftarrow \alpha^{t-1}/2$
      `NonImprCtr` $\leftarrow 0$
    else
      $\alpha^t \leftarrow \alpha^{t-1}$
    end if
  end if
  get feasible solution from Algorithm 8.2 with input $\bar{x}$ ▷ Upper bound
  x, y, $z(x, y) \leftarrow$ output of Algorithm 8.2
  if $z(x, y) <$ UB **then** ▷ Compare to best-known upper bound
    UB $\leftarrow z(x, y)$
    $(x^{\text{UB}}, y^{\text{UB}}) \leftarrow (x, y)$
  end if
  $\Delta^t \leftarrow \alpha^t(\text{UB} - z_{\text{LR}}(\lambda^t)) / \sum_{i \in I} \left(1 - \sum_{j \in J} \bar{y}_{ij}\right)^2$ ▷ Update multipliers
  for all $i \in I$ **do**
    $\lambda_i^{t+1} \leftarrow \lambda_i^t + \Delta^t \left(1 - \sum_{j \in J} \bar{y}_{ij}\right)$
  end for
  $t \leftarrow t + 1$ ▷ Increment t
until UB $- z_{\text{LR}}(\lambda^t) \leq \kappa$ or $t > \zeta$ ▷ Check for termination
return x^{UB}, y^{UB}, UB

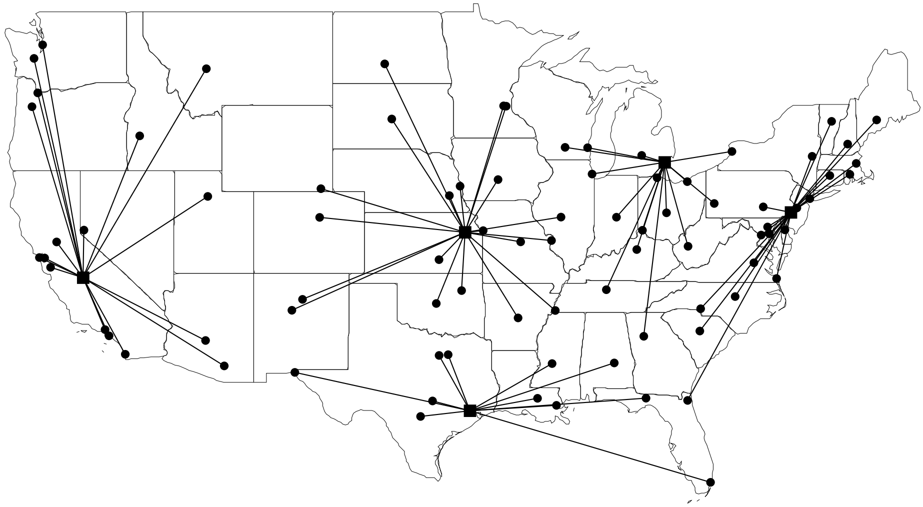


Figure 8.2 Optimal solution to 88-node UFLP instance. Total cost = \$783,813.

3-facility solution in Figure 8.1(b) has a total cost of \$1,238,911 (\$512,800 fixed cost plus \$726,111 transportation cost).

□

8.2.3.7 Variable Fixing Sometimes the Lagrangian relaxation procedure terminates with the lower and upper bounds farther apart than we'd like. Before executing branch-and-bound to close the gap, we may be able to fix some of the x_j variables to 0 or 1 based on the facility benefits and the current bounds. The variables can be fixed permanently, throughout the entire branch-and-bound tree. The more variables we can fix, the faster the branch-and-bound procedure is likely to run. Essentially, the method works by "peeking" down a branch of the tree and running a quick check to determine whether the next node down the branch would be fathomed.

Theorem 8.3 *Let UB be the best upper bound found during the Lagrangian procedure, let λ be a given set of Lagrange multipliers that were used during the procedure, let β_j be the facility benefits* (8.13) *under λ, and let $z_{\text{LR}}(\lambda)$ be the lower bound (the optimal objective value of* (UFLP-LR$_\lambda$)*) under λ. If $x_j = 0$ in the solution to* (UFLP-LR$_\lambda$) *and*

$$z_{\text{LR}}(\lambda) + \beta_j + f_j > \text{UB}, \tag{8.23}$$

then $x_j = 0$ in every optimal solution to (UFLP). *If $x_j = 1$ in the solution to* (UFLP-LR$_\lambda$) *and*

$$z_{\text{LR}}(\lambda) - (\beta_j + f_j) > \text{UB}, \tag{8.24}$$

then $x_j = 1$ in every optimal solution to (UFLP).

Proof. Suppose we were to branch on x_j, setting $x_j = 0$ for one child node and $x_j = 1$ for the other, and suppose we use λ as the initial multipliers for the Lagrangian procedure at each child node.

At the "$x_j = 1$" node, the same facilities would be open as in the root-node solution, except that now facility j is also open. The cost of this solution for (UFLP-LR$_\lambda$) is the cost of the original solution, $z_{\text{LR}}(\lambda)$, plus $\beta_j + f_j$. Therefore, we would obtain $z_{\text{LR}}(\lambda) + \beta_j + f_j$ as a lower bound at this node. Since this lower bound is greater than the best-known upper bound, we would fathom the tree at this node, and the optimal solution would be contained in the other half of the tree—the "$x_j = 0$" half.

A similar argument applies to the second case. At the "$x_j = 0$" node, we obtain a lower bound of $z_{\text{LR}}(\lambda) - (\beta_j + f_j)$, and if this is greater than UB, we fathom the tree at this node. ■

Note that, in the second part of the theorem, if $x_j = 1$ then, by (8.14), $\beta_j + f_j < 0$, which is why the left-hand side of (8.24) might be greater than UB.

This trick has been applied successfully to a variety of facility location problems; see, e.g., Daskin et al. (2002) and Snyder and Daskin (2005). Typically, the conditions in Theorem 8.3 are checked twice after processing has terminated at the root node, once using the most recent multipliers λ and once using the multipliers that produced the best-known lower bound. The time required to check these conditions for every j is negligible.

8.2.3.8 Alternate Relaxation As stated above, we could have chosen instead to relax constraints (8.5). In this case, the Lagrangian subproblem becomes

$$\begin{aligned}
(\text{UFLP-LR}_\lambda) \quad \text{minimize} \quad & \sum_{j\in J} f_j x_j + \sum_{i\in I}\sum_{j\in J} h_i c_{ij} y_{ij} + \sum_{i\in I}\sum_{j\in J} \lambda_{ij}\left(x_j - y_{ij}\right) \\
&= \sum_{j\in J}\left(\sum_{i\in I}\lambda_{ij} + f_j\right) x_j + \sum_{i\in I}\sum_{j\in J}(h_i c_{ij} - \lambda_{ij}) y_{ij} \qquad (8.25) \\
\text{subject to} \quad & \sum_{j\in J} y_{ij} = 1 \qquad \forall i \in I \qquad (8.26) \\
& x_j \in \{0,1\} \qquad \forall j \in J \qquad (8.27) \\
& y_{ij} \geq 0 \qquad \forall i \in I, \forall j \in J \qquad (8.28)
\end{aligned}$$

Now every customer must be assigned to a single facility, but that facility need not be open. There are no constraints linking the x and y variables, so the problem can be written as two separate problems:

$$\begin{aligned}
(x\text{-problem}) \quad \text{minimize} \quad & \sum_{j\in J}\left(\sum_{i\in I}\lambda_{ij} + f_j\right) x_j \qquad (8.29) \\
\text{subject to} \quad & x_j \in \{0,1\} \qquad \forall j \in J \qquad (8.30)
\end{aligned}$$

$$\begin{aligned}
(y\text{-problem}) \quad \text{minimize} \quad & \sum_{i\in I}\sum_{j\in J}(h_i c_{ij} - \lambda_{ij}) y_{ij} \qquad (8.31) \\
\text{subject to} \quad & \sum_{j\in J} y_{ij} = 1 \qquad \forall i \in I \qquad (8.32) \\
& y_{ij} \geq 0 \qquad \forall i \in I, \forall j \in J \qquad (8.33)
\end{aligned}$$

To solve the x-problem, we simply set $x_j = 1$ for all j such that $\sum_{i\in I} \lambda_{ij} + f_j < 0$. (Note that since the constraints relaxed are $\le$ constraints, $\lambda \le 0$; see Section D.1.5.1.) To solve the y-problem, for each i, we set $y_{ij} = 1$ for the j that minimizes $h_i c_{ij} - \lambda_{ij}$. The rest of the procedure is similar, except that the step-size calculation becomes

$$\Delta^t = \frac{\alpha^t(\mathrm{UB} - z_{\mathrm{LR}}(\lambda^t))}{\sum_{i\in I}\sum_{j\in J}(x_j - y_{ij})^2} \tag{8.34}$$

and the multiplier-updating formula becomes

$$\lambda_{ij}^{t+1} = \lambda_{ij}^t + \Delta^t(x_j - y_{ij}). \tag{8.35}$$

In practice, relaxing the assignment constraints (8.4) tends to work better than relaxing the linking constraints (8.5). One reason for this is that the former relaxation involves relaxing fewer constraints, which generally makes it easier to find good multipliers using subgradient optimization. Another reason is that since y_{ij} will be 0 for many j that are open, there will be many constraints such that $y_{ij} < x_j$. It is often difficult to get good results when relaxing inequality constraints if many of them have slack.

8.2.4 The DUALOC Algorithm

The DUALOC algorithm was proposed by Erlenkotter (1978). It is a *dual-ascent* or *primal–dual* algorithm that constructs good feasible solutions for the dual of the LP relaxation of (UFLP) and then uses these to develop good (often optimal) integer solutions for the primal, i.e., for (UFLP) itself.

We form the LP relaxation of (UFLP), denoted (UFLP-P), by replacing constraints (8.6) with

$$x_j \ge 0 \qquad \forall j \in J. \tag{8.36}$$

Let v_i and w_{ij} be the dual variables for constraints (8.4) and (8.5), respectively. In addition, for notational convenience, let $\hat{c}_{ij} \equiv h_i c_{ij}$. Then the LP dual is given by

$$\text{(UFLP-D)} \quad \text{maximize} \quad \sum_{i\in I} v_i \tag{8.37}$$

$$\text{subject to} \quad \sum_{i\in I} w_{ij} \le f_j \qquad \forall j \in J \tag{8.38}$$

$$v_i - w_{ij} \le \hat{c}_{ij} \qquad \forall i \in I, \forall j \in J \tag{8.39}$$

$$w_{ij} \ge 0 \qquad \forall i \in I, \forall j \in J \tag{8.40}$$

The complementary slackness conditions for (UFLP-P) and (UFLP-D) are given by

$$x_j^*\left(f_j - \sum_{i\in I} w_{ij}^*\right) = 0 \tag{8.41}$$

$$y_{ij}^*\left[\hat{c}_{ij} - (v_i^* - w_{ij}^*)\right] = 0 \tag{8.42}$$

$$v_i^*\left(1 - \sum_{j\in J} y_{ij}^*\right) = 0 \tag{8.43}$$

$$w_{ij}^* \left(y_{ij}^* - x_j^*\right) = 0 \tag{8.44}$$

Suppose we are given arbitrary values of the variables v_i. Then either it is feasible to set w_{ij} as small as possible, i.e.,

$$w_{ij} = \max\{0, v_i - \hat{c}_{ij}\} \tag{8.45}$$

for all i and j, or there are *no* feasible w_{ij} values (for the given v_i values). Moreover, since w_{ij} does not appear in the objective function, any feasible w_{ij} (for fixed v_i) is acceptable. Thus, we assume that (8.45) holds and substitute this relationship into (UFLP-D). Constraints (8.39) and (8.40) are automatically satisfied when (8.45) holds; therefore, we obtain the following *condensed dual*, which uses only v_i and not w_{ij}:

$$\text{(UFLP-CD)} \quad \text{maximize} \quad \sum_{i \in I} v_i \tag{8.46}$$

$$\text{subject to} \quad \sum_{i \in I} \max\{0, v_i - \hat{c}_{ij}\} \le f_j \qquad \forall j \in J \tag{8.47}$$

Substituting (8.45) into the complementary slackness conditions (8.41)–(8.44), we obtain

$$x_j^* \left(f_j - \sum_{i \in I} \max\{0, v_i^* - \hat{c}_{ij}\} \right) = 0 \tag{8.48}$$

$$y_{ij}^* \left[\hat{c}_{ij} - (v_i^* - \max\{0, v_i^* - \hat{c}_{ij}\})\right] = 0 \tag{8.49}$$

$$v_i^* \left(1 - \sum_{j \in J} y_{ij}^* \right) = 0 \tag{8.50}$$

$$\max\{0, v_i^* - \hat{c}_{ij}\} \left(y_{ij}^* - x_j^*\right) = 0 \tag{8.51}$$

Note that (UFLP-CD) is not an LP, since the $\max\{\cdot\}$ function is nonlinear. One could develop a customized simplex-type algorithm to solve it—an approach like this is proposed by Schrage (1978) for the p-median problem, among others—but instead, the DUALOC approach exploits the structure of (UFLP-CD) to find near-optimal solutions directly.

The DUALOC algorithm consists of two main procedures. The first is a *dual-ascent procedure* that generates feasible dual solutions for (UFLP-CD) and corresponding primal integer solutions for (UFLP). The second is a *dual-adjustment procedure* that attempts to reduce complementary slackness violations (thereby improving the primal or dual solutions, or both) by adjusting the dual solution iteratively and calling the dual-ascent procedure as a subroutine. If these procedures terminate without an optimal integer solution to (UFLP), branch-and-bound is used to close the optimality gap.

8.2.4.1 Primal–Dual Relationships The dual-ascent procedure generates both a dual solution v^+ for (UFLP-CD) and a set $J^+ \subseteq J$ of facility locations such that the following properties hold:

- *Primal–Dual Property 1* (PDP1): $\sum_{i \in I} \max\{0, v_i^+ - \hat{c}_{ij}\} = f_j$ for all $j \in J^+$
- *Primal–Dual Property 2* (PDP2): For each $i \in I$, there exists at least one $j \in J^+$ such that $\hat{c}_{ij} \le v_i^+$

Such a solution can easily be converted to an integer primal solution: The set J^+ provides the x variables for (UFLP), and, as in the Lagrangian relaxation procedure (Section 8.2.3.4), the y variables can be set by assigning each customer to its nearest open facility. That is, an integer primal solution for (UFLP) can be obtained from J^+ as follows:

$$x_j^+ = \begin{cases} 1, & \text{if } j \in J^+ \\ 0, & \text{otherwise} \end{cases} \tag{8.52}$$

$$y_{ij}^+ = \begin{cases} 1, & \text{if } j = j^+(i) \\ 0, & \text{otherwise,} \end{cases} \tag{8.53}$$

where $j^+(i) \equiv \text{argmin}_{k \in J^+}\{\hat{c}_{ik}\}$.

The primal–dual solution (x^+, y^+, v^+) satisfies three of the four complementary slackness conditions: (8.48) is satisfied because of PDP1, and (8.50) is satisfied because each i is assigned to exactly one j in (8.53). To see why (8.49) is satisfied, suppose $y_{ij}^+ = 1$, i.e., $j = j^+(i)$. By PDP2, $\hat{c}_{ij} \le v_i^+$ for some $j \in J^+$ and $\hat{c}_{i,j^+(i)} \le \hat{c}_{ij}$ by the definition of $j^+(i)$, so

$$\begin{aligned} & y_{i,j^+(i)}^+ \left[\hat{c}_{i,j^+(i)} - \left(v_i^+ - \max\{0, v_i^+ - \hat{c}_{i,j^+(i)}\}\right)\right] \\ =& \hat{c}_{i,j^+(i)} - \left(v_i^+ - (v_i^+ - \hat{c}_{i,j^+(i)})\right) \\ =& 0. \end{aligned}$$

Thus, (x^+, y^+) and v^+ are optimal for (UFLP-P) and (UFLP-CD), respectively, if and only if (8.51) holds. Moreover, since (x^+, y^+) is integer, if it is optimal for (UFLP-P), then it is also optimal for (UFLP). (It may seem strange to hope that the integer solution (x^+, y^+) is optimal for the LP relaxation (UFLP-P). But remember that (UFLP-P) often has all-integer solutions "for free" (see page 272), and is usually very tight when it is not all-integer so that good integer solutions to (UFLP-P) are likely to be good also for (UFLP).)

Condition (8.51) may be violated when $\hat{c}_{ij} < v_i^+$ for some $j \ne j^+(i)$, since in that case $y_{ij}^+ = 0$ but $x_j^+ = 1$. This suggests that complementary slackness violations can be reduced by focusing on the $v_i^+ - \hat{c}_{ij}$ terms for $j \ne j^+(i)$, and indeed those terms directly affect the duality gap, as the next lemma attests.

Lemma 8.4 *Let z_P^+ be the objective function value of (UFLP-P) under the solution (x^+, y^+), and let z_D^+ be the objective function value of (UFLP-CD) under the solution v^+. Then*

$$z_P^+ - z_D^+ = \sum_{i \in I} \sum_{\substack{j \in J^+ \\ j \ne j^+(i)}} \max\{0, v_i^+ - \hat{c}_{ij}\}.$$

Proof. Omitted; see Problem 8.40. ■

The dual-ascent procedure (Section 8.2.4.2) generates v^+ and J^+. The dual-adjustment procedure (Section 8.2.4.3) then attempts to improve the solutions by reducing $v_i^+ - \hat{c}_{ij}$ terms for $j \ne j^+(i)$.

8.2.4.2 The Dual-Ascent Procedure The dual-ascent procedure constructs a dual solution v^+ and a facility set J^+ such that properties PDP1 and PDP2 hold for v^+ and J^+. The procedure begins by constructing an easy feasible solution in which the v_i variables are small (in order to ensure feasibility with respect to (8.47)) and then increasing the v_i one by one (in order to improve the objective (8.46)).

For each $i \in I$, sort the costs $\hat{c}_{ij}$ in nondecreasing order and let $\hat{c}_i^k$ be the kth of these costs, for $k = 1, \ldots, |J|$. Define $\hat{c}_i^{|J|+1} \equiv \infty$. Then an initial solution can be generated by setting $v_i = \hat{c}_i^1$ for all $i \in I$; this solution is feasible for (UFLP-CD). (Why?) Actually, any initial feasible solution will work, but this one is easy to obtain.

The dual-ascent procedure is given in Algorithm 8.4. In line 1, we initialize v_i to $\hat{c}_i^1$ and initialize the index k_i to 2. Throughout the algorithm, k_i equals the smallest k such that $v_i \lneqq \hat{c}_i^k$; as the v_i increase in the algorithm, so do the k_i. In line 2, s_j represents the slack in constraint (8.47) for facility j; since v_i equals the smallest $\hat{c}_{ij}$, $s_j = f_j - \sum_{i \in I} \max\{0, v_i - \hat{c}_{ij}\} = f_j$. The algorithm loops through the customers; for each customer i, we would like to set v_i to the next larger value of $\hat{c}_{ij}$, i.e., to $\hat{c}_i^{k_i}$. However, increasing v_i increases the left-hand side of (8.47) for all j such that $v_i - \hat{c}_{ij} \geq 0$. (These j are the facilities whose costs are $\hat{c}_i^1, \ldots, \hat{c}_i^{k_i - 1}$.) Therefore, line 6 calculates the largest allowable increase in v_i without violating (8.47) for any j. Note that we only consider j such that $v_i - \hat{c}_{ij} \geq 0$; for the other j, the left-hand sides of (8.47) will not increase because we will not increase v_i past $\hat{c}_i^{k_i}$, as enforced by lines 7–8. Lines (9)–(10) update the `IMPROVED` flag and the index k_i. (The `IMPROVED` flag is only set to `TRUE` if we were able to increase v_i all the way to $\hat{c}_i^{k_i}$ for some i, not for smaller increases.) Lines 12–14 adjust the slack for all facilities whose left-hand sides of (8.47) will change, and line 15 performs the update to v_i. The process repeats until v_i cannot be increased to $\hat{c}_i^{k_i}$ for any customer. Line 18 sets v^+ equal to the final value of v and builds the set J^+, and the algorithm returns both these values.

Algorithm 8.4 Dual-ascent procedure for DUALOC algorithm

1: $v_i \leftarrow \hat{c}_i^1,\ k_i \leftarrow 2\ \forall i \in I$ ▷ Initialization
2: $s_j \leftarrow f_j\ \forall j \in J$
3: **repeat** ▷ Improvement
4:     `IMPROVED` $\leftarrow$ `FALSE`
5:     **for all** $i \in I$ **do**
6:         $\Delta_i \leftarrow \min_{j \in J: v_i - \hat{c}_{ij} \geq 0}\{s_j\}$ ▷ Calculate allowable increase in v_i
7:         **if** $\Delta_i \geq \hat{c}_i^{k_i} - v_i$ **then** ▷ Did we get all the way to $\hat{c}_i^{k_i}$?
8:             $\Delta_i \leftarrow \hat{c}_i^{k_i} - v_i$
9:             `IMPROVED` $\leftarrow$ `TRUE`
10:             $k_i \leftarrow k_i + 1$
11:         **end if**
12:         **for all** $j \in J$ s.t. $v_i - \hat{c}_{ij} \geq 0$ **do**
13:             $s_j \leftarrow s_j - \Delta_i$ ▷ Adjust slack
14:         **end for**
15:         $v_i \leftarrow v_i + \Delta_i$ ▷ Adjust v_i
16:     **end for**
17: **until** not `IMPROVED` ▷ Stop when no improvement
18: $v^+ \leftarrow v,\ J^+ \leftarrow \{j \in J | s_j = 0\}$ ▷ Build solutions to return
19: **return** v^+, J^+

Proposition 8.5 *The v^+ and J^+ returned by Algorithm 8.4 satisfy PDP1 and PDP2.*

Proof. PDP1: It suffices to show that, throughout the progression of the algorithm, $s_j = f_j - \sum_{i \in I} \max\{0, v_i - \hat{c}_{ij}\}$. (We use v to refer to the values set during the course of the algorithm, and v^+ to refer to the final values returned by the algorithm.) Clearly, this holds after line 2. In the main loop, each time v_i increases by Δ_i for any i, then either $v_i - \hat{c}_{ij} \geq 0$, in which case we reduce s_j by Δ_i; or $v_i - \hat{c}_{ij} < 0$, in which case we increase v_i to at most $\hat{c}_{ij}$ (in line 8), so $f_j - \max\{0, v_i - \hat{c}_{ij}\}$ does not change, and neither does s_j. In other words, at the end of each iteration through the main loop, $s_j = f_j - \sum_{i \in I} \max\{0, v_i - \hat{c}_{ij}\}$.

PDP2: Suppose, for a contradiction, that there exists an $i \in I$ such that $\hat{c}_{ij} > v_i^+$ for all $j \in J^+$. This means that $s_j > 0$ for all j such that $v_i^+ - \hat{c}_{ij} \geq 0$. Then at line 6, Δ_i would have been set to a positive number, and at line 15, v_i would have been increased by Δ_i. This contradicts our assumption that v^+ is the solution returned by the algorithm. ■

If there is a strict subset of J^+ that still satisfies PDP1 and PDP2, it is better to use that subset. To see why, suppose there is a facility j' with $s_{j'} = 0$ that is not included in J^+. PDP1 does not prohibit this situation; it prohibits the converse. Would it be better to add j' to J^+? Lemma 8.4 suggests the answer is no: For each $i \in I$, either $\hat{c}_{ij'} < \hat{c}_{i,j^+(i)}$ (so j' becomes the new closest facility to i), in which case z_P^+ increases by $v_i^+ - \hat{c}_{i,j^+(i)} > 0$; or $\hat{c}_{ij'} \geq \hat{c}_{i,j^+(i)}$, in which case z_P^+ increases by $\max\{0, v_i - \hat{c}_{ij'}\} \geq 0$. Therefore, we want J^+ to be *minimal* in the sense that no facility can be removed from it without violating PDP2. Of course, finding a minimal J^+ is itself a combinatorial problem. Erlenkotter (1978) suggests a simple heuristic for finding such a set, but to keep things simple, we'll just assume that J^+ contains *all* j for which $s_j = 0$.

You might be wondering why we limit Δ_i to $\hat{c}_i^{k_i} - v_i$ in line 8, since we want v_i to be as large as possible, and we can leave Δ_i at the value set in line 6 while maintaining feasibility. Recall that the complementary slackness condition (8.51) is violated when $\hat{c}_{ij} < v_i$ and j is open but i is not assigned to j. There tend to be fewer of these violations when we spread the $\hat{c}_{ij} < v_i$ among the customers i rather than having a few customers with very large v_i values.

Once we have J^+, we can generate an integer primal solution (x^+, y^+) using (8.52) and (8.53). If (x^+, y^+, v^+) satisfies (8.51) for all i and j, then the complementary slackness conditions are all satisfied and (x^+, y^+) is optimal. If, instead, (8.51) is violated for some i and j, then we attempt to reduce these violations using the dual-adjustment procedure, described in the next section.

☐ **EXAMPLE 8.2**

Figure 8.3 depicts an instance of the UFLP with four customers (marked as circles) and three potential facility sites (marked as squares). Fixed costs f_j are marked next to each facility. Each customer has a demand of $h_i = 1$, and transportation costs are equal to the Manhattan-metric distance between the facility and customer. We will use DUALOC's dual-ascent procedure (Algorithm 8.4) to find a feasible solution for this instance.

First, we sort the transportation costs for each customer. In Figure 8.4, for each customer, each facility is positioned based on its distance from the customer. The

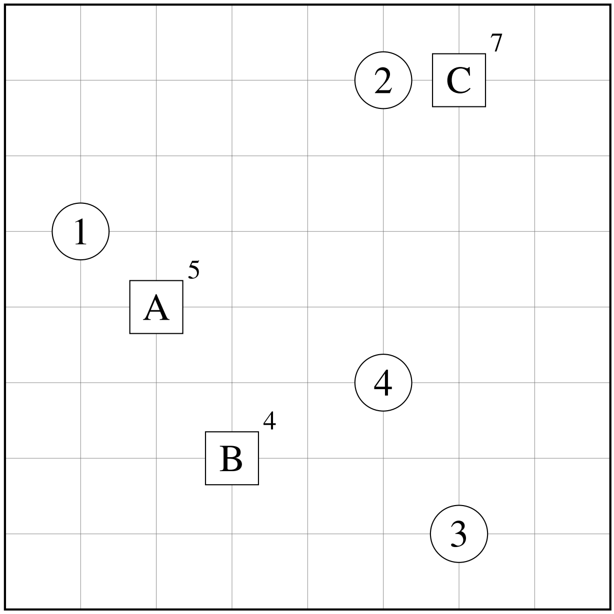


Figure 8.3 Customer and facility layout for Example 8.2.

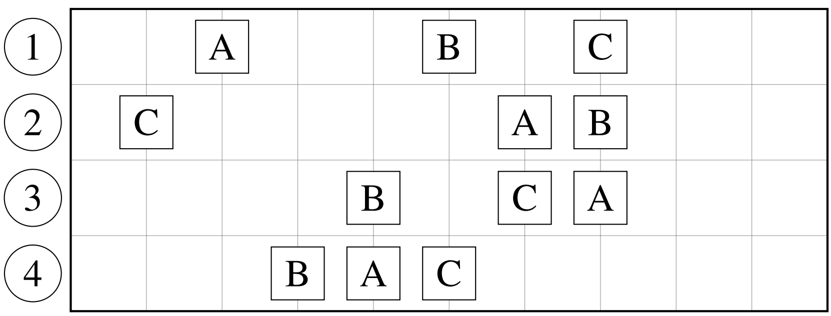


Figure 8.4 Sorted facility positions for Example 8.2.

algorithm begins by setting $v_i \leftarrow \hat{c}_i^1$ and $k_i = 2$ for all i, and $s_j \leftarrow f_j$ for all j:

$$\begin{array}{lll} v_1 = \hat{c}_{1\mathrm{A}} = 2 & k_1 = 2 & s_\mathrm{A} = 5 \\ v_2 = \hat{c}_{2\mathrm{C}} = 1 & k_2 = 2 & s_\mathrm{B} = 4 \\ v_3 = \hat{c}_{3\mathrm{B}} = 4 & k_3 = 2 & s_\mathrm{C} = 7 \\ v_4 = \hat{c}_{4\mathrm{B}} = 3 & k_4 = 2 & \end{array}$$

(You can imagine v_i as being positioned at the relevant spot in Figure 8.4.) Next the algorithm searches for v_i values to increase. One could choose to loop through the customers in any order; we'll go in increasing order to keep things simple. For $i = 1$, facility A is the only j for which $v_1 - \hat{c}_{1j} \geq 0$, so we have $\Delta_1 \leftarrow s_\mathrm{A} = 5$. Since $\Delta_1 > \hat{c}_{1\mathrm{B}} - v_1 = 3$, we set $\Delta_1 \leftarrow 3$, $k_1 \leftarrow 3$, $s_\mathrm{A} \leftarrow 2$, and $v_1 \leftarrow 5$. Similarly, for $i = 2$, $\Delta_2 \leftarrow s_\mathrm{C} = 7$; since $\Delta_2 > \hat{c}_{2\mathrm{C}} - v_2 = 5$, we set $\Delta_2 \leftarrow 5$, $k_2 \leftarrow 3$, $s_\mathrm{C} \leftarrow 2$, and $v_2 \leftarrow 6$. For $i = 3$: $\Delta_3 \leftarrow 2$, $k_3 \leftarrow 3$, $s_\mathrm{B} \leftarrow 2$, and $v_3 \leftarrow 6$. Finally, for $i = 4$: $\Delta_4 \leftarrow 1$, $k_4 \leftarrow 3$, $s_\mathrm{B} \leftarrow 1$, and $v_4 \leftarrow 4$. At the end of this iteration, we have:

$$\begin{array}{lll} v_1 = 5 & k_1 = 3 & s_\mathrm{A} = 2 \\ v_2 = 6 & k_2 = 3 & s_\mathrm{B} = 1 \\ v_3 = 6 & k_3 = 3 & s_\mathrm{C} = 2 \\ v_4 = 4 & k_4 = 3 & \end{array}$$

Since Δ_i reached $\hat{c}_i^{k_i} - v_i$ for at least one i (actually, for all of them), we repeat for another iteration. For $i = 1$: $\Delta_1 \leftarrow \min\{s_\mathrm{A}, s_\mathrm{B}\} = 1$ since $v_1 - \hat{c}_{1j} \geq 0$ for both $j =$ A and B. Since $\Delta_1 < \hat{c}_1^{k_1} - v_1$, we leave Δ_1 and k_1 where they are, and then set $s_\mathrm{A} \leftarrow 1$, $s_\mathrm{B} \leftarrow 0$, and $v_1 \leftarrow 6$. For $i = 2$: $\Delta_2 \leftarrow \min\{s_\mathrm{A}, s_\mathrm{C}\} = 1$ since $v_2 - \hat{c}_{2j}$ for both $j =$ A and C. Since $\Delta_2 = \hat{c}_2^{k_2} - v_2$, we set $k_2 \leftarrow 4$, $s_\mathrm{A} \leftarrow 0$, $s_\mathrm{C} \leftarrow 1$, and $v_2 \leftarrow 7$. For $i = 3$ and 4, we have $\Delta_3,\ \Delta_4 \leftarrow 0$ (since $s_\mathrm{B} = 0$), so there is nothing to do. At the end of this iteration, we have:

$$\begin{array}{lll} v_1 = 6 & k_1 = 3 & s_\mathrm{A} = 0 \\ v_2 = 7 & k_2 = 4 & s_\mathrm{B} = 0 \\ v_3 = 6 & k_3 = 3 & s_\mathrm{C} = 1 \\ v_4 = 4 & k_4 = 3 & \end{array}$$

Since Δ_2 reached $\hat{c}_2^{k_2} - v_2$, we repeat for another iteration. However, at this iteration, we cannot increase v_i for any i since $s_\mathrm{A} = s_\mathrm{B} = 0$, so the **repeat** $\cdots$ **until** loop terminates.

The algorithm returns $v^+ = (6, 7, 6, 4)$ and $J^+ = \{\mathrm{A}, \mathrm{B}\}$. The feasible primal solution obtained from (8.52)–(8.53) is $x^+ = (1, 1, 0)$ and $y^+_{1\mathrm{A}} = y^+_{2\mathrm{A}} = y^+_{3\mathrm{B}} = y^+_{4\mathrm{B}} = 1$. The dual solution v^+ has objective value $z_D^+ = 23$ and the primal solution has objective value $z_P^+ = 24$. The fact that there is a duality gap indicates that *either* we have not found an optimal solution to the dual LP, *or* we have not found an optimal solution to the primal IP, *or* there is an integrality gap, i.e., the LP relaxation has a fractional optimum.

We can't tell which—yet. To resolve this question, we would run the dual-adjustment procedure. For this instance, the dual-adjustment procedure would yield no improvement to the solution above. We would then use branch-and-bound to close the duality gap, and we would find that there is an optimal integer solution in which we locate only at facility B and to assign all customers to it, for a total cost of 23.

Therefore, the dual solution found by the dual-ascent procedure was optimal, but the corresponding primal solution was not. □

8.2.4.3 The Dual-Adjustment Procedure The dual-adjustment procedure identifies customers and facilities that violate the complementary slackness condition (8.51) and reduces these violations by decreasing the dual variable v_i for some $i \in I$. Doing so frees up slack on some of the binding constraints (8.47), which allows us to increase $v_{i'}$ for other $i' \in I$. Each unit of decrease in v_i allows one unit of increase in $v_{i'}$ (since the coefficients in (8.51) equal 1). The dual objective value will increase if more than one $v_{i'}$ can be increased in this way and will stay the same if only one can be increased. In either case, we may obtain a new (potentially better) primal solution since the set J^+ might change.

We face three questions: (1) Which dual variables v_i are candidates for reduction? (2) Once we reduce v_i, adding slack to some of the constraints, which $v_{i'}$ are candidates for increase? (3) How much should we increase each of the candidate $v_{i'}$? We'll answer each of these questions in turn.

Which v_i to decrease? A customer i is a candidate for reduction in v_i if it violates the complementary slackness condition (8.51). The next lemma characterizes those customers in terms of v and $\hat{c}$.

Lemma 8.6 *Let v be a dual solution and J^+ be a facility set that satisfy PDP1 and PDP2, and let (x^+, y^+) be the corresponding feasible solution calculated from* (8.52)–(8.53). *Then $i \in I$ violates* (8.51) *if and only if $v_i > \hat{c}_{ij}$ for at least two $j \in J^+$.*

Proof. $i \in I$ violates (8.51) if and only if there is some $j' \in J^+$ such that $v_i > \hat{c}_{ij'}$ but $y^+_{ij'} = 0$. This happens if and only if i is assigned to a different $j'' \in J^+$, i.e., if and only if there is a $j'' \in J^+$ such that $\hat{c}_{ij''} \le \hat{c}_{ij'}$. This happens if and only if $v_i \ge \hat{c}_{ij}$ for at least two $j \in J^+$. ■

Therefore, a customer i is a candidate for reduction in v_i if $v_i > \hat{c}_{ij}$ for at least two $j \in J^+$. The algorithm reduces it only as far as the next-smaller $\hat{c}_{ij}$; that is, it reduces it to $\hat{c}_i^-$, where $\hat{c}_i^-$ is the largest $\hat{c}_{ij}$ (among all $j \in J$) that is strictly less than v_i:

$$\hat{c}_i^- = \max_{j \in J} \{\hat{c}_{ij} | v_i > c_{ij}\}.$$

Which $v_{i'}$ to increase? Suppose $v_i > \hat{c}_{ij}$ for at least two $j \in J^+$ and so we reduce v_i. This adds slack to (8.47) for all $j \in J$ such that $v_i > \hat{c}_{ij}$. Lemma 8.6 implies that two of these constraints correspond to $j^+(i)$ and $j^{++}(i)$, where $j^{++}(i)$ is the second-closest facility in J^+ to i. (Recall that $j^+(i)$ is the closest.) Suppose there is some $i' \in I$ for which there is only one $j \in J^+$ such that $v_{i'} \ge \hat{c}_{i'j}$. We'll say that i' is *solely constrained* by j in this case, because j is the only facility preventing an increase in $v_{i'}$. If i' is solely constrained by $j^+(i)$ or $j^{++}(i)$, then a decrease in v_i can be matched by an increase $v_{i'}$. The algorithm therefore focuses on such i', attempting to increase their $v_{i'}$ values first. It also uses the "solely constrained" test to identify candidates for reduction in v_i: If there are no i' that are solely constrained by $j^+(i)$ or $j^{++}(i)$, the algorithm does not consider reducing v_i, even if i is a candidate for a decrease in v_i as described above.

How much to increase $v_{i'}$? Deciding which $v_{i'}$ to increase, and by how much, is precisely the intent of the dual-ascent procedure! Therefore, the dual-adjustment procedure uses the dual-ascent procedure as a subroutine—first with the set of customers restricted to the candidates for increases in $v_{i'}$, then with i added, and then with the full customer set I.

The dual-adjustment procedure is described in pseudocode in Algorithm 8.5. The algorithm loops through the customers to identify candidates for reducing v_i. A customer is a candidate if (1) it violates the complementary slackness condition (8.51) (this check occurs in line 3, making use of Lemma 8.6), and (2) there are at least two customers that are solely constrained by either $j^+(i)$ or $j^{++}(i)$ (this check occurs in lines 4–5). Assuming customer i passes both checks, lines 6–8 increase the slack for all j for which $v_i > \hat{c}_{ij}$, and line 9 reduces v_i to the next smallest $\hat{c}_{ij}$ value.

Next, the algorithm calls the dual-ascent procedure to decide how to use up the newly created slack. Line 10 restricts I to the customers that are solely constrained by $j^+(i)$ or $j^{++}(i)$; line 11 adds i itself to this set; and line 12 runs the dual-ascent algorithm on the entire set I in order to ensure a valid solution v^+. If v_i has increased, the adjustment procedure repeats (for the same i), and this continues until there is no improvement or v_i reaches its original value. At that point, we move on to the next customer. The algorithm terminates when all customers have been considered.

Algorithm 8.5 Dual-adjustment procedure for DUALOC algorithm

1: **for all** $i \in I$ **do** ▷ Loop through customers
2:     **repeat**
3:         **if** $v_i > \hat{c}_{ij}$ for at least two $j \in J^+$ **then** ▷ Check for CSC violation
4:             $I^+ \leftarrow \{i' \in I | i'$ is solely constrained by $j^+(i)$ or $j^{++}(i)\}$
5:             **if** $I^+ \neq \emptyset$ **then** ▷ Are there other $v_{i'}$ we can increase?
6:                 **for all** $j \in J$ s.t. $v_i > \hat{c}_{ij}$ **do**
7:                     $s_j \leftarrow s_j + v_i - \hat{c}_i^-$ ▷ Increase slack
8:                 **end for**
9:                 $v_i \leftarrow \hat{c}_i^-$ ▷ Reduce v_i
10:                 Run Alg. 8.4 with I restricted to I^+ ▷ Increase other $v_{i'}$
11:                 Run Alg. 8.4 with I restricted to $I^+ \cup \{i\}$
12:                 Run Alg. 8.4 for full I
13:             **end if**
14:         **end if**
15:     **until** no improvement in dual objective or v_i has resumed its original value
16: **end for**

If the dual-ascent and dual-adjustment procedures result in primal and dual solutions (v^+, x^+, y^+) whose objective values are equal, then v^+ is optimal for the dual LP and (x^+, y^+) is optimal for the primal LP *and* IP. If the objectives are unequal, then a straightforward implementation of branch-and-bound can be used to close the optimality gap. Erlenkotter (1978) reports excellent computational results for this method on several test problems, typically with little or no branching required.

Körkel (1989) proposes computational improvements that speed the DUALOC algorithm up considerably. DUALOC has been adapted to solve many other problems, such as the p-median problem discussed in Section 8.3.2 (Galvão 1980, Nauss and Markland 1981), the stochastic UFLP discussed in Section 8.6.2 (Mirchandani et al. 1985), the Steiner tree problem (Wong 1984), and general supply chain network design problems (Balakrishnan et al. 1989). Goemans and Williamson (1997) discuss DUALOC and other primal–dual algorithms.

8.2.5 Heuristics for the UFLP

Heuristics for combinatorial problems such as the UFLP fall into two categories: *construction heuristics* and *improvement heuristics*. Construction heuristics build a feasible solution from scratch, whereas improvement heuristics start with a feasible solution and attempt to improve it.

The most basic construction heuristics for the UFLP are *greedy* heuristics such as the "greedy-add" procedure (Kuehn and Hamburger 1963): Start with all facilities closed and open the single facility that can serve all customers with the smallest objective function value; then at each iteration open the facility that gives the largest decrease in the objective, stopping when no facility can be opened that will decrease the objective. (See Algorithm 8.6. In the algorithm, x^k, y^k, and z^k refer to the solution when facility k is (temporarily) opened.)

Algorithm 8.6 Greedy-add heuristic for UFLP

$x_j \leftarrow 0 \; \forall \, j \in J$; $z \leftarrow \infty$ ▷ Initialization
repeat
  IMPROVED $\leftarrow$ FALSE
  for all $k \in J$ s.t. $x_k = 0$ **do** ▷ Main loop
    $x^k \leftarrow x$, $y^k \leftarrow y$ ▷ Make copy of current solution
    $x^k_k \leftarrow 1$ ▷ Open facility k
    for all $i \in I$ **do**
      $j(i) \leftarrow \text{argmin}_{j \in J : x^k_j = 1}\{c_{ij}\}$ ▷ Assign i to nearest open j
      $y^k_{i,j(i)} \leftarrow 1$
    end for
    $z^k \leftarrow \sum_{j\in J} f_j x^k_j + \sum_{i \in I} \sum_{j \in J} h_i c_{ij} y^k_{ij}$ ▷ Calculate cost if open j
  end for
  if $\min_{k\in J}\{z^k\} < z$ **then** ▷ Compare to current cost
    $k^* \leftarrow \text{argmin}_{k \in J}\{z^k\}$ ▷ Open best facility
    $x_{k^*} \leftarrow 1$; $y \leftarrow y^{k^*}$; $z \leftarrow \min_{k \in J}\{z^k\}$ ▷ Update current solution
    IMPROVED $\leftarrow$ TRUE
  end if
until not IMPROVED
return x, y, z

☐ **EXAMPLE 8.3**

Let us apply the greedy-add heuristic to the UFLP instance in Example 8.1. We begin with all facilities closed and, one by one, calculate the cost of opening each facility and assigning all customers to it. For example, if we open facility 1 (New York, NY), the single-facility solution costs \$2,591,762 (\$189,600 in fixed cost and \$2,402,162 in transportation cost). If we open facility 2 (Los Angeles, CA), the cost is \$3,638,252, and so on. The best and worst facilities to open, given that we only open one facility, sorted by cost, are listed in Table 8.1, and a few of the corresponding

Table 8.1 Greedy algorithm costs: Iteration 1.

Rank	Facility #	City, State	Cost
1	34	St. Louis, MO	1,935,714
2	69	Springfield, IL	1,941,395
3	13	Indianapolis, IN	1,950,055
4	80	Jefferson City, MO	1,970,644
5	44	Cincinnati, OH	1,983,877
		⋮	
84	68	Salem, OR	3,911,294
85	11	San Jose, CA	3,913,554
86	81	Olympia, WA	3,991,424
87	14	San Francisco, CA	4,019,984
88	21	Seattle, WA	4,030,326

Table 8.2 Greedy algorithm costs: Iteration 2.

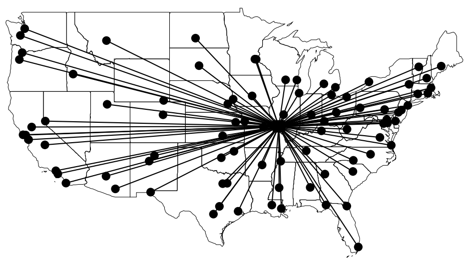
Rank	Facility #	City, State	Cost
1	46	Fresno, CA	1,338,962
2	9	Phoenix, AZ	1,391,766
3	78	Carson City, NV	1,398,997
4	41	Sacramento, CA	1,414,305
5	33	Tucson, AZ	1,424,947
		⋮	
83	42	Minneapolis, MN	1,930,728
84	51	St. Paul, MN	1,933,291
85	69	Springfield, IL	1,936,401
86	66	Tallahassee, FL	1,945,300
87	45	Miami, FL	1,980,578

solutions are depicted in Figure 8.5. Since St. Louis, MO is the best city to open, we open it and leave it open for the duration of the heuristic.

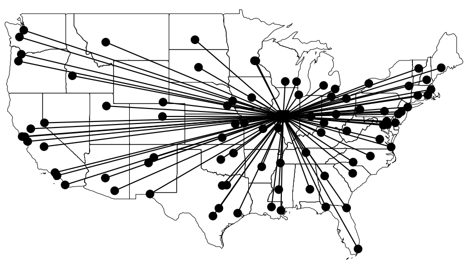
Next we determine the best second facility to open, given that St. Louis is also open. The best and worst facilities are given in Table 8.2 and Figure 8.6.

So, we fix open the facilities in St. Louis, MO, and Fresno, CA. Proceeding in this manner, in iteration 3, we open facility 5, in Philadelphia, PA, to obtain the solution shown in Figure 8.7(a), which has a cost of 904,055. In iteration 4, we open facility 28 in Fort Worth, TX, for a solution with a cost of 821,501 (Figure 8.7(b)). In iteration 5, we open facility 7 in Detroit, MI, for a solution with a cost of 793,443 (Figure 8.7(c)). In iteration 6, the best facility to open is facility 15 (Jacksonville, FL), but the resulting solution has a cost of 807,938, which is greater than the cost of the previous solution. Therefore, the heuristic terminates, returning the 5-facility solution in Figure 8.7(c).

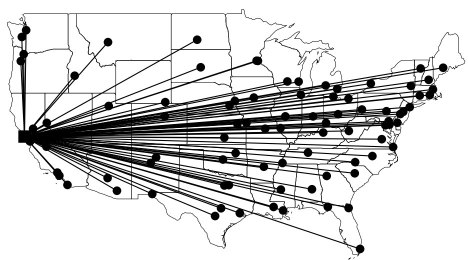
□



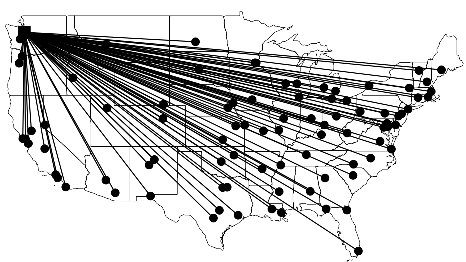
(a) St. Louis, MO (cost = 1,935,714).



(b) Springfield, IL (cost = 1,941,395).

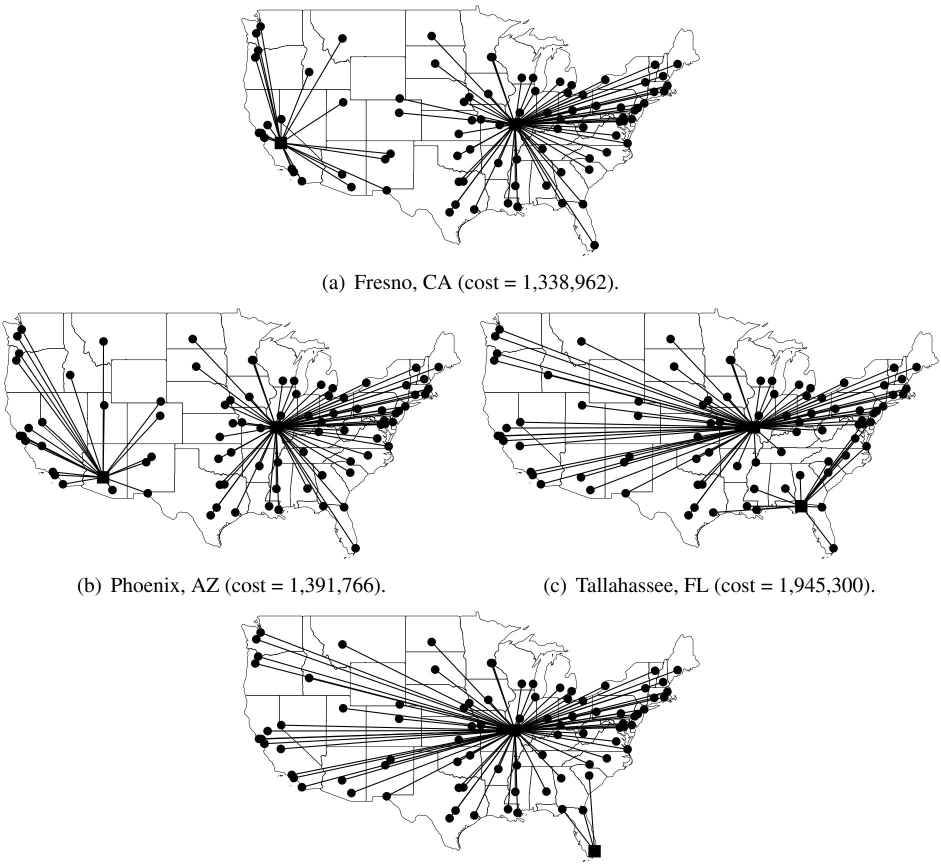


(c) San Francisco, CA (cost = 4,019,984).



(d) Seattle, WA (cost = 4,030,326).

Figure 8.5 Considering each facility for iteration 1 of greedy algorithm for UFLP.



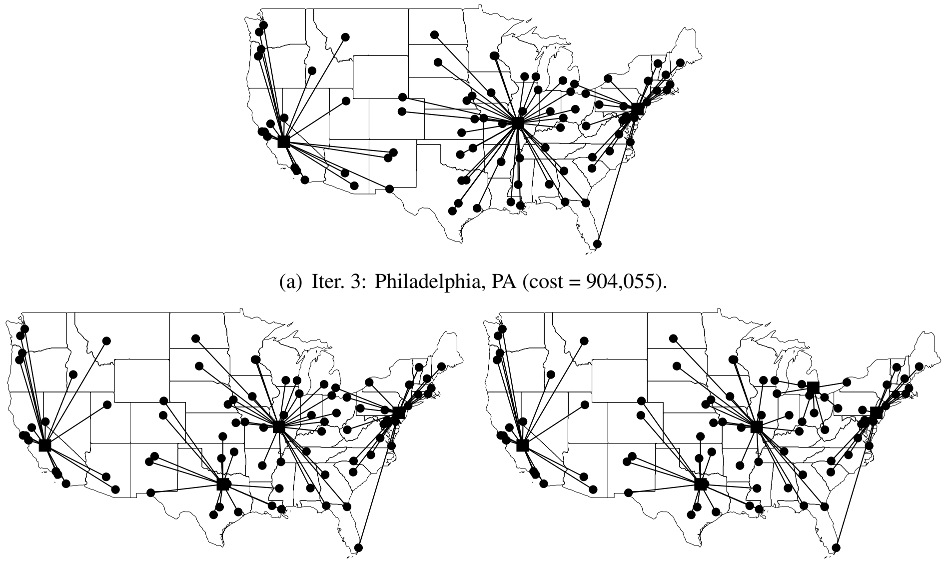
(a) Fresno, CA (cost = 1,338,962).

(b) Phoenix, AZ (cost = 1,391,766).

(c) Tallahassee, FL (cost = 1,945,300).

(d) Miami, FL (cost = 1,980,578).

Figure 8.6 Considering each facility for iteration 2 of greedy algorithm for UFLP, with facility in St. Louis, MO, fixed open.



(a) Iter. 3: Philadelphia, PA (cost = 904,055).

(b) Iter. 4: Fort Worth, TX (cost = 821,501). (c) Iter. 5: Detroit, MI (cost = 793,443).

Figure 8.7 Solutions from iterations 3, 4, and 5 of greedy algorithm for UFLP.

By assuming that the next facility to open will be the last, the greedy-add heuristic can easily fall into a trap. For example, if it is optimal to open two facilities, the greedy-add heuristic may first open a facility in the center of the geographical region, which then must stay open for the second iteration, when in fact it is optimal to open one facility on each side of the region.

A reverse approach is called the "greedy-drop" heuristic, which starts with all facilities open and sequentially closes the facility that decreases the objective the most. It has similar advantages and disadvantages as greedy-add.

One important improvement heuristic is the *swap* or *exchange* heuristic (Teitz and Bart 1968), which attempts to find a facility to open and a facility to close such that the new solution has a smaller objective function value. For more on the swap heuristic, see Section 8.3.2.3. Other procedures attempt to find closed facilities that can be opened to reduce the objective function, or open facilities that can be closed.

The heuristics mentioned here have proven to perform well in practice, which means they return good solutions *and* execute quickly. Metaheuristics have also been widely applied to the UFLP. These include genetic algorithms (Jaramillo et al. 2002), tabu search (Al-Sultan and Al-Fawzan 1999), and simulated annealing (Arostegui et al. 2006).

8.3 OTHER MINISUM MODELS

The UFLP is an example of a *minisum* location problem. Minisum models are so called because their objective is to minimize a sum of the costs or distances between customers and their assigned facilities (as well as possibly other terms, such as fixed costs). In contrast, *covering* location problems are more concerned with the maximum distance, with the goal of ensuring that most or all customers are located close to their assigned facilities.

At the risk of over-generalizing, it can be said that minisum models are more commonly applied in the private sector, in which profits and costs are the dominant concerns, and covering models are most commonly applied in the public sector, in which service, fairness, and equity are more important. For further discussion of this dichotomy, see Revelle et al. (1970).

In this section, we discuss two other minisum models—the capacitated fixed-charge location problem and the p-median problem. In Section 8.4, we discuss covering models.

8.3.1 The Capacitated Fixed-Charge Location Problem (CFLP)

In the UFLP, we assumed that there are no capacity restrictions on the facilities. Obviously, this is an unrealistic assumption in many practical settings. The UFLP can be easily modified to account for capacity restrictions; the resulting problem (not surprisingly) is called the *capacitated fixed-charge location problem*, or CFLP. Suppose v_j is the maximum demand that can be served by facility j per year. The CFLP can be formulated as follows:

$$\text{(CFLP)} \quad \text{minimize} \quad \sum_{j\in J} f_j x_j + \sum_{i\in I}\sum_{j\in J} h_i c_{ij} y_{ij} \tag{8.54}$$

$$\text{subject to} \quad \sum_{j\in J} y_{ij} = 1 \qquad \forall i \in I \tag{8.55}$$

$$y_{ij} \le x_j \qquad \forall i \in I, \forall j \in J \tag{8.56}$$

$$\sum_{i\in I} h_i y_{ij} \le v_j \qquad \forall j \in J \tag{8.57}$$

$$x_j \in \{0,1\} \qquad \forall j \in J \tag{8.58}$$

$$y_{ij} \ge 0 \qquad \forall i \in I, \forall j \in J \tag{8.59}$$

This IP is identical to (UFLP) except for the new capacity constraints (8.57). Sometimes the following constraint is added, which says that the total capacity of the opened facilities is sufficient to meet the total demand:

$$\sum_{j\in J} v_j x_j \ge \sum_{i\in I} h_i. \tag{8.60}$$

This constraint is redundant in the IP formulation but tightens some relaxations.

Many approaches have been proposed to solve this problem. We briefly outline a method very similar to the method discussed for the UFLP. We relax the assignment constraints (8.55) to obtain the following Lagrangian subproblem:

$$\begin{aligned}\text{(CFLP-LR}_\lambda) \quad \text{minimize} \quad & \sum_{j\in J} f_j x_j + \sum_{i\in I}\sum_{j\in J} h_i c_{ij} y_{ij} + \sum_{i\in I} \lambda_i \left(1 - \sum_{j\in J} y_{ij}\right) \\ = & \sum_{j\in J} f_j x_j + \sum_{i\in I}\sum_{j\in J} (h_i c_{ij} - \lambda_i) y_{ij} + \sum_{i\in I} \lambda_i \end{aligned} \tag{8.61}$$

$$\text{subject to} \quad y_{ij} \le x_j \qquad \forall i \in I, \forall j \in J \tag{8.62}$$

$$\sum_{i\in I} h_i y_{ij} \le v_j \qquad \forall j \in J \tag{8.63}$$

$$x_j \in \{0,1\} \qquad \forall j \in J \tag{8.64}$$
$$y_{ij} \geq 0 \qquad \forall i \in I, \forall j \in J \tag{8.65}$$

As in the UFLP, this problem separates by j, but now computing the benefit β_j is a little more complicated because of the capacity constraint. In particular, for each $j \in J$, we need to solve a problem of the form

$$(\mathrm{P}_j) \quad \text{minimize} \quad \beta_j = \sum_{i \in I} a_i z_i \tag{8.66}$$
$$\text{subject to} \quad \sum_{i \in I} h_i z_i \leq v \tag{8.67}$$
$$0 \leq z_i \leq 1 \qquad \forall i \in I, \tag{8.68}$$

where $a_i = h_i c_{ij} - \lambda_i$, $z_i = y_{ij}$, and $v = v_j$. This is a continuous knapsack problem, which can be solved efficiently by sorting the is so that

$$\frac{a_1}{h_1} \leq \frac{a_2}{h_2} \leq \cdots \leq \frac{a_{|I|}}{h_{|I|}}.$$

(This sort order favors large negative values of a_i and small positive values of h_i.) We then set $z_i = 1$ for $i = 1, \ldots, r$, where r is the largest number such that $a_r < 0$ and

$$\sum_{i=1}^{r} h_i \leq v_j.$$

If $r < |I|$, we set $z_{r+1} = \left(v_j - \sum_{i=1}^{r} h_i\right)/h_{r+1}$. Other aspects of the Lagrangian procedure (finding upper bounds, subgradient optimization, branch-and-bound) are similar to those discussed in Section 8.2.3, although the upper-bounding procedure must take into account the capacity constraints.

Several other relaxations for the CFLP have been studied, often using slightly different formulations from (CFLP). Davis and Ray (1969) solve the LP relaxation of the CFLP in a branch-and-bound algorithm, as do Akinc and Khumawala (1977). Nauss (1978) and Christofides and Beasley (1983) use Lagrangian relaxation, relaxing constraints (8.55), similar to the method outlined above. Klincewicz and Luss (1986) relax the capacity constraints (8.57) to obtain a UFLP. Van Roy (1986) also relaxes (8.57) but rather than using standard Lagrangian relaxation, he uses cross-decomposition, a hybrid of Lagrangian relaxation and Benders decomposition. Barcelo et al. (1991) use variable splitting (Guignard and Kim 1987), also known as Lagrangian decomposition, a method in which some of the variables are doubled, the new variables are forced equal to the original ones via a constraint, and that constraint is then relaxed using Lagrangian relaxation. Also see Geoffrion and McBride (1978) and Cornuejols et al. (1991) for a discussion of the relative tightness of the theoretical bounds from the various relaxations of the CFLP.

Generally, the optimal solution to (CFLP) will not have $y_{ij} \in \{0,1\}$ as in (UFLP). (Why?) This means that some customers will receive product from more than one DC. Sometimes it is important to prohibit this from happening by requiring $y_{ij} \in \{0,1\}$; this is called a *single-sourcing constraint*. The CFLP with single-sourcing constraints is harder to solve because (P_j) becomes a 0–1 knapsack problem, which is NP-hard. On the other hand, good algorithms exist for the knapsack problem, and since the knapsack problem does not have the integrality property, the Lagrangian bound will be tighter than the LP bound. This highlights the important trade-off between the quality of the Lagrangian bound and the ease with which the subproblem can be solved.

A closely related problem is the *capacitated concentrator location problem* (CCLP), in which the demands h_i are ignored in the objective function (or, equivalently, the transportation costs c_{ij} are divided by h_i) but not in the capacity constraints. The CCLP features prominently in the location-based heuristic for the vehicle routing problem (Section 11.3.3). See Mirzaian (1985), Klincewicz and Luss (1986), and Gourdin et al. (2002).

8.3.2 The p-Median Problem (pMP)

In the UFLP, the fixed costs in the objective function prevent the model from opening too many facilities. Another way to accomplish the same thing is simply to add a constraint that explicitly limits the number of open facilities. This is the approach taken by the *p-median problem* (pMP), which was introduced by Hakimi (1965).

Hakimi focused on problems on networks, in which the distances among nodes are defined not by a geographical measure like Euclidean or great circle distances, but rather on shortest-path distances along the edges of the network. (See Section 8.2.2.) His main result, which has come to be known as the *Hakimi property*, is that there is always an optimal solution consisting of nodes of the network rather than points along the edges. In particular, suppose $I = J$ are the nodes of the network. For any set X consisting of p points on the network (either at the nodes or along the edges) and for any $i \in I$, define $c(i, X)$ to be the shortest-path distance from i to the nearest point in X. (This is a generalization of the notation c_{ij} to consider distances to points that are not nodes.) Hakimi proved the following:

Theorem 8.7 (Hakimi (1965)) *There exists a set $I_p^* \subseteq I$ consisting of p nodes of the network such that, for any set X consisting of p points on the network (nodes or edges),*

$$\sum_{i \in I} h_i c(i, I_p^*) \le \sum_{i \in I} h_i c(i, X).$$

In other words, there exists an optimal set that consists only of nodes. This allows us to treat the problem as a discrete one consisting of a finite number of feasible solutions rather than a continuous one with an infinite number. Hakimi solved the pMP using complete enumeration of all subsets of p nodes, but of course this approach only works for small p and $|I|$. Many more efficient algorithms have been proposed since Hakimi's original work, several of which we discuss below.

8.3.2.1 Formulation The pMP uses the same notation as the UFLP (Section 8.2.2), plus the following:

Parameter

p = number of facilities to locate

The problem is formulated as follows:

$$(p\text{MP}) \quad \text{minimize} \quad \sum_{i \in I} \sum_{j \in J} h_i c_{ij} y_{ij} \tag{8.69}$$

$$
\begin{aligned}
\text{subject to} \quad & \sum_{j \in J} y_{ij} = 1 && \forall i \in I && (8.70)\\
& y_{ij} \le x_j && \forall i \in I, \forall j \in J && (8.71)\\
& \sum_{j \in J} x_j = p && && (8.72)\\
& x_j \in \{0,1\} && \forall j \in J && (8.73)\\
& y_{ij} \ge 0 && \forall i \in I, \forall j \in J && (8.74)
\end{aligned}
$$

The objective function (8.69) computes the transportation cost. (Often c_{ij} is defined as distance, rather than per-unit cost, in which case the objective function is interpreted as representing the *demand-weighted distance*.) Constraint (8.72) requires exactly p facilities to be opened. This constraint could be written with a $\le$ instead of $=$, but since the objective function decreases with the number of open facilities, the optimal solution under such a constraint will always open exactly p facilities; therefore, the two forms of the constraint are equivalent. The other constraints function the same as the corresponding constraints in the (UFLP).

8.3.2.2 Exact Algorithms for the pMP The pMP is NP-hard for arbitrary p but is polynomially solvable if p is fixed (Garey and Johnson 1979). This means that there exist algorithms for which the worst-case running time is a polynomial function of the problem size ($|I|$, $|J|$) but not of p. The pMP can also be solved in polynomial time for arbitrary p when the underlying network is a tree (Kariv and Hakimi 1979b), i.e., when the distance matrix is derived from shortest-path distances on a tree network. Despite its NP-hardness, however, the pMP, like the UFLP, can be solved relatively efficiently, partly due to the fact that its LP relaxation is typically quite tight (ReVelle and Swain 1970).

The Lagrangian relaxation procedure discussed in Section 8.2.3 can be easily modified for the pMP (Cornuejols et al. 1977). Relaxing constraints (8.70), we obtain the following Lagrangian subproblem:

$$
\begin{aligned}
(p\text{MP-LR}_\lambda) \quad \text{minimize} \quad & \sum_{i \in I} \sum_{j \in J} h_i c_{ij} y_{ij} + \sum_{i \in I} \lambda_i \left(1 - \sum_{j \in J} y_{ij}\right)\\
&= \sum_{i \in I} \sum_{j \in J} (h_i c_{ij} - \lambda_i) y_{ij} + \sum_{i \in I} \lambda_i && && (8.75)\\
\text{subject to} \quad & y_{ij} \le x_j && \forall i \in I, \forall j \in J && (8.76)\\
& \sum_{j \in J} x_j = p && && (8.77)\\
& x_j \in \{0,1\} && \forall j \in J && (8.78)\\
& y_{ij} \ge 0 && \forall i \in I, \forall j \in J && (8.79)
\end{aligned}
$$

The benefit of opening facility j is given by (8.13), exactly as in the UFLP. Then, we set $x_j = 1$ for the p facilities with the smallest β_j (negative or positive). (Recall that for the UFLP, we set $x_j = 1$ if and only if $\beta_j + f_j < 0$.) Finally, we set $y_{ij} = 1$ if $x_j = 1$ and $h_i c_{ij} - \lambda_i < 0$. The optimal objective function value of the subproblem,

$$
z_{\text{LR}}(\lambda) = \sum_{j \in J} \beta_j x_j + \sum_{i \in I} \lambda_i,
$$

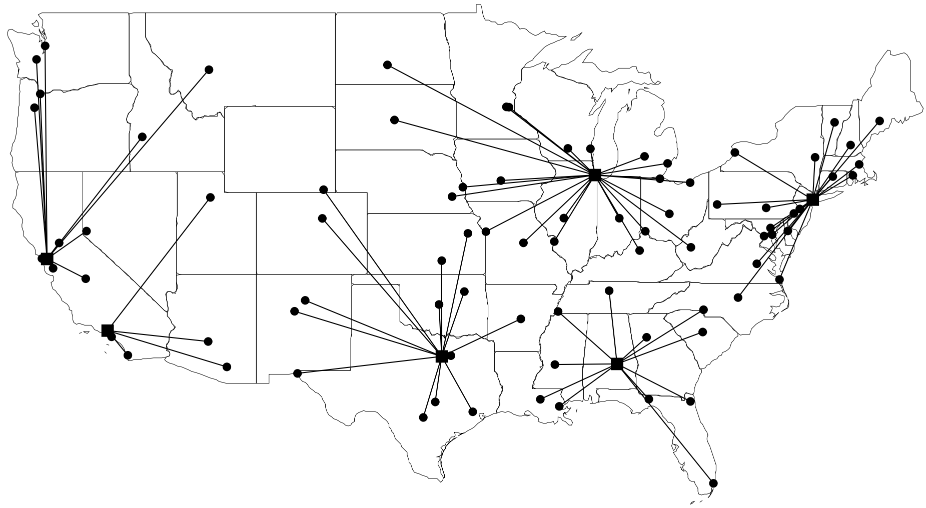


Figure 8.8 Optimal solution to 88-node pMP instance with $p = 6$. Total cost = \$386,856.

provides a lower bound on the optimal objective function value of (pMP). Feasible solutions can be found by assigning each customer to the nearest facility that is open in the solution to the subproblem, and the corresponding objective function value provides an upper bound. The Lagrange multipliers can be found using subgradient optimization (Section 8.2.3.5). Branch-and-bound can then be used to close any remaining optimality gap.

☐ **EXAMPLE 8.4**

Return to the 88-node data set described in Example 8.1. The optimal solution to the pMP with $p = 6$ is shown in Figure 8.8. This solution locates facilities in New York, NY; Los Angeles, CA; Chicago, IL; Fort Worth, TX; Oakland, CA; and Montgomery, AL, with a total cost of \$386,856. As in Example 8.1, MATLAB took less than 2 seconds to solve this problem to optimality on a laptop computer.

☐

Other exact methods include LP relaxation/branch-and-bound (ReVelle and Swain 1970), decomposition methods (Garfinkel et al. 1974), row and column reduction (Rosing et al. 1979), and adaptations by Galvão (1980) and Nauss and Markland (1981) of the DUALOC algorithm (Erlenkotter 1978) discussed in Section 8.2.4. Reese (2006) provides a thorough survey of the literature on the pMP, including both exact and heuristic algorithms.

8.3.2.3 Heuristics for the pMP Most heuristics for the UFLP (Section 8.2.5) are easily adapted for the pMP. For instance, we can apply the greedy-add and greedy-drop heuristics, except that the procedure terminates when there are exactly p facilities open rather than when no objective-reducing adds or drops can be found.

One of the earliest and most widely known heuristics for the pMP is the *swap* or *exchange* heuristic introduced by Teitz and Bart (1968). The swap heuristic attempts to find a pair j, k of facilities with j open and k closed such that if j were closed and k opened (and the customers reassigned as needed), the objective function value would decrease. If such a pair can be found, the swap is made and the procedure continues. Pseudocode for the swap heuristic is given in Algorithm 8.7. It takes as inputs the current solution variable x and its objective value z. In the pseudocode, $(\bar{x}, \bar{y})$ is a temporary solution and $\bar{z}$ is its cost.

Algorithm 8.7 Swap heuristic for pMP

input current solution x, current cost z
repeat ▷ Main loop
  IMPROVED $\leftarrow$ FALSE
  for all $j \in J$ s.t. $x_j = 1$ **do** ▷ Loop through open facilities
    for all $k \in J$ s.t. $x_k = 0$ **do** ▷ Loop through closed facilities
      $\bar{x} \leftarrow x$; $\bar{x}_j \leftarrow 0$; $\bar{x}_k \leftarrow 1$ ▷ Try swapping j and k
      for all $i \in I$ **do**
        $j(i) \leftarrow \text{argmin}_{j \in J: \bar{x}_j = 1}\{c_{ij}\}$ ▷ Assign i to nearest open j
        $\bar{y}_{i,j(i)} \leftarrow 1$; $\bar{y}_{i,\ell} \leftarrow 0 \; \forall \ell \in J \setminus \{j(i)\}$
      end for
      $\bar{z} \leftarrow \sum_{i \in I} \sum_{\ell \in J} h_i c_{i\ell} \bar{y}_{i\ell}$ ▷ Calculate cost if swap j and k
      if $\bar{z} < z$ **then** ▷ Check for improvement
        $x \leftarrow \bar{x}$; $y \leftarrow \bar{y}$, $z \leftarrow \bar{z}$ ▷ Update current solution
        IMPROVED $\leftarrow$ TRUE
      end if
    end for
  end for
until not IMPROVED
return x, y, z

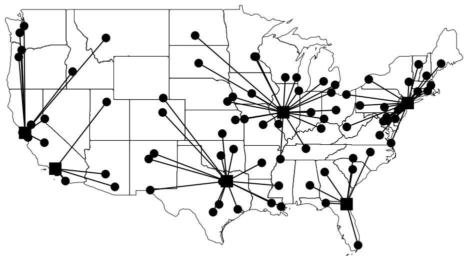
□ **EXAMPLE 8.5**

Applying the greedy-add heuristic to the 88-node instance described in Example 8.4, we open the following facilities, in sequence: Springfield, IL; Los Angeles, CA; New York, NY; Dallas, TX; Jacksonville, FL; Oakland, CA. The resulting solution, shown in Figure 8.9(a), has a cost of \$423,620—9.5% more expensive than the optimal solution found in Example 8.4.

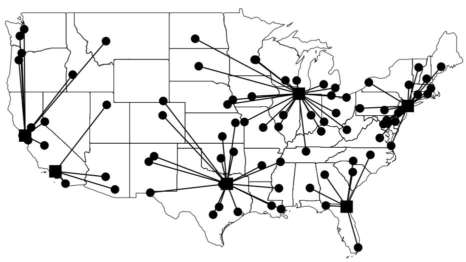
Let us now apply the swap heuristic to the greedy solution. First, we can close the facility in Springfield, IL, and open the one in Chicago, IL, to reduce the cost by 7.6%, to \$391,314. Next, we can close Jacksonville, FL, and open Atlanta, GA, for a new cost of \$387,226; then close Dallas, TX, in favor of Ft. Worth, TX (\$387,021); and finally close Atlanta, GA (opened a few iterations earlier) in favor of Montgomery, AL (\$386,856). These moves are shown in Figures 8.9(b)–8.9(e). No other profitable swaps can be made, and in fact, this is the optimal solution found in Example 8.4. Note, however, that the greedy and swap heuristics do not find the optimal solution in all instances—we just got lucky for this one.

□

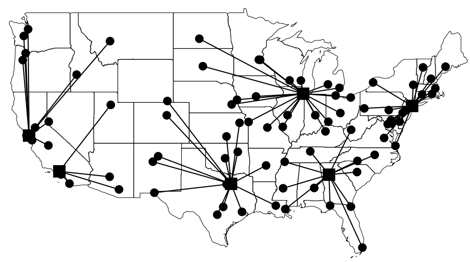
The swap heuristic can be modified in many ways. For example, at each iteration, we can either implement the first swap we find that reduces the objective (this is called a *first-improving strategy*) or implement the swap that reduces the objective function the



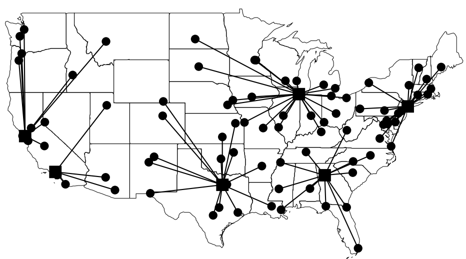
(a) Greedy-add solution (cost = \$423,620).



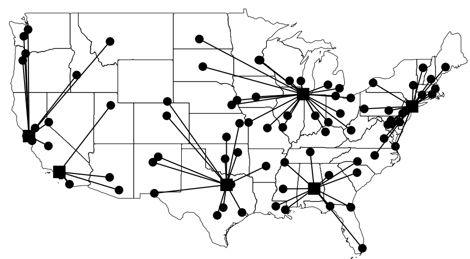
(b) Swap iter. 1: close Springfield, open Chicago (cost = \$391,314).



(c) Swap iter. 2: close Jacksonville, open Atlanta (cost = \$387,226).



(d) Swap iter. 3: close Dallas, open Ft. Worth (cost = \$387,021).



(e) Swap iter. 4: close Atlanta, open Montgomery (cost = \$386,856).

Figure 8.9 Greedy and swap solutions for 88-node pMP instance with $p = 6$.

most (a *best-improving strategy*). (Algorithm 8.7 uses a first-improving strategy.) Or, we can *randomize* the procedure by choosing randomly from among, for example, the five best swaps at each iteration, possibly with a bias toward the better swaps.

A straightforward implementation of the swap heuristic is relatively slow since we must evaluate $O(p|J|)$ possible swaps and, for each, determine $O(|I|)$ customer assignments, for an overall complexity of $O(p|I||J|)$ for each iteration. Whitaker (1983) proposed an implementation known as *fast interchange*, which was further refined by Hansen and Mladenović (1997) so that each iteration takes $O(|I|(|J|-p))$.

Another improvement heuristic is the *neighborhood search* heuristic (Maranzana 1964). For simplicity, assume that $I = J$, that is, every node is both a customer and a potential facility location. Define the *neighborhood* of an open facility j in a given solution, denoted N_j, as the set of nodes i that are assigned to j. The neighborhood search heuristic solves the 1-median problem in each neighborhood N_j to check whether j is in fact the best facility for N_j. If it is not, it closes j and opens the 1-median. The neighborhoods are then redefined (i.e., the customers are reallocated), and the procedure repeats. Pseudocode for the neighborhood search heuristic is given in Algorithm 8.8.

Algorithm 8.8 Neighborhood search heuristic for pMP

input current solution x, y
repeat ▷ Main loop
    `IMPROVED` $\leftarrow$ `FALSE`
    for all $j \in J$ s.t. $x_j = 1$ **do** ▷ Loop through open facilities
        $N_j \leftarrow \{i \in I : y_{ij} = 1\}$ ▷ Determine neighborhood of j
        $k \leftarrow \mathrm{argmin}_{\ell \in N_j} \left\{ \sum_{i \in N_j} h_i c_{i\ell} \right\}$ ▷ Determine 1-median of N_j
        if $k \neq j$ **then**
            $x_j \leftarrow 0$; $x_k \leftarrow 1$ ▷ Swap j and k
            $j(i) \leftarrow \mathrm{argmin}_{j \in J: x_j = 1} \{c_{ij}\}$ ▷ Assign i to nearest open j
            $y_{i,j(i)} \leftarrow 1$; $y_{i,\ell} \leftarrow 0 \ \forall \ell \in J \setminus \{j(i)\}$
            `IMPROVED` $\leftarrow$ `TRUE`
        end if
    end for
until not `IMPROVED`
$z \leftarrow \sum_{i \in I} \sum_{\ell \in J} h_i c_{i\ell} y_{i\ell}$ ▷ Calculate new cost
return x, y, z

The neighborhood search heuristic is, in some ways, similar to the swap heuristic in the sense that it searches for an open facility j to close and a closed facility k to open. The difference is that the neighborhood search heuristic only searches over facilities k that are in j's neighborhood, and when it evaluates the new cost after swapping j and k, it only considers reassignments of customers in the neighborhood, rather than the entire customer set. Both of these differences lead to some loss of accuracy, but also significantly faster run times. (See Problem 8.45.)

The discussion above assumed that $I = J$. If $I \neq J$, then instead of searching over the neighborhood of j, N_j, for a new facility k, we must instead define some suitable set M_j of facilities that are likely candidates for the 1-median of N_j. For example, we might set M_j to the set of $k \in J$ that are in the convex hull of the points in N_j. The pseudocode in

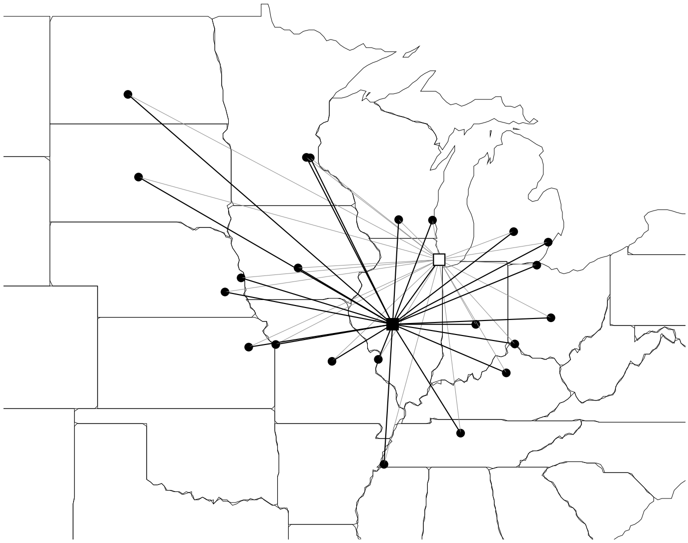


Figure 8.10 Neighborhood of Springfield, IL in greedy solution to 88-node instance for pMP with $p = 6$.

Algorithm 8.8 remains the same except for step 6, in which we would replace $\ell \in N_j$ with $\ell \in M_j$.

☐ **EXAMPLE 8.6**

Consider the solution found by the greedy-add heuristic in Example 8.5, shown in Figure 8.9(a). The neighborhood of the facility in Springfield, IL, is shown in Figure 8.10. It happens that Springfield is *not* the 1-median for this neighborhood; Chicago is. (Chicago is shown in the figure as an open square; light lines connect the customers to that facility.) So, we close Springfield and open Chicago, then check whether there are any Springfield customers that should be assigned to a facility *other* than Chicago, or any non-Springfield customers that should now be assigned to Chicago. (There are not.) Making this swap reduces the cost to \$391,314.

The other open facilities *are* the 1-medians of their respective neighborhoods, so there are no more moves to make, and the heuristic terminates.

☐

Many metaheuristics are available for the pMP. For example, Hosage and Goodchild (1986) propose a genetic algorithm (GA) for the pMP, one of the first GAs for facility location problems. Chiyoshi and Galvão (2000) propose a simulated annealing algorithm for the pMP that makes use of the swap heuristic. Hansen and Mladenović (1997) propose a variable neighborhood search (VNS) heuristic. For a survey of metaheuristic approaches for the pMP, see Mladenovic et al. (2007).

Figure 8.11 400-mile coverage radii around facilities in 6-median solution to 88-node instance. Total covered demand is 4268 out of 3979 (88.7%). Solid customers are covered by an open facility; shaded customers are not. Radii around different facilities have different sizes due to the Mercator projection used in this map, which exaggerates distances farther from the equator.

8.4 COVERING MODELS

In 2001, the National Fire Protection Association established Standard 1710, which, among many other guidelines, says that fire departments should have the objective of arriving to a fire within 4 minutes of receiving a call (National Fire Protection Association, Inc. 2001). The ability of a fire department to adhere to this standard is driven largely by the locations of its fire stations, since a fire will surely have to wait more than 4 minutes if it is located too far from its nearest fire station, no matter how quickly the firefighters respond.

However, this is not an objective that the UFLP, pMP, or other minisum models can help much with, since the optimal solutions to those problems may assign some customers to very distant facilities if it is cost effective to do so. Instead, we need to use the notion of *coverage*, which indicates whether a given customer is within a prespecified distance, or *coverage radius*, of an open facility.

For example, Figure 8.11 shows the optimal facilities from the 6-median problem on the 88-node data set (from Figure 8.8), along with 400-mile coverage radii around each facility. (Since transportation costs c_{ij} for this data set are equal to 0.5 times the distance, a 400-mile coverage radius is the same as a \$200 coverage radius.) Some customers are covered, but many are not, especially in the western part of the United States, which is more sparsely populated (and hence less expensive to serve with long hauls in minisum models). In total, the covered nodes have a demand of 3979, or 88.7% of the total demand of 4484.

Note that in Figure 8.11 and others in this section, radii around different facilities are drawn in different sizes due to the Mercator projection used in this map, which exaggerates distances farther from the equator.

In this section, we discuss three seminal facility location models that use coverage to determine the quality of the solution. The first, the *set covering location problem* (SCLP), locates the minimum number of facilities to cover every demand node. The second, the *maximal covering location problem* (MCLP), covers as many demands as possible while locating a fixed number of facilities. In other words, the SCLP puts the number of facilities in the objective function while constraining the coverage, and the MCLP does the reverse. The third model, the *p-center problem*, locates a fixed number of facilities to minimize the maximum distance from a demand node to its nearest open facility—or, put another way, to minimize the coverage radius required to cover every demand node.

For further reading on covering problems, see Snyder (2011) or Daskin (2013).

8.4.1 The Set Covering Location Problem (SCLP)

In the *set covering location problem* (SCLP), we are required to cover *every* demand node; the objective is to do so with the fewest possible number of facilities. The SCLP was first formulated in a facility location context by Hakimi (1965), though similar models appeared in graph-theoretic settings prior to that.

In addition to the notation introduced in earlier sections, we use the following new notation:

Parameters

a_{ij} = 1 if facility $j \in J$ can cover customer $i \in I$ (if it is open), 0 otherwise

The coverage parameter a_{ij} can be derived from a distance or cost parameter such as c_{ij} in the UFLP, for example:

$$a_{ij} = \begin{cases} 1, & \text{if } c_{ij} \le r \\ 0, & \text{otherwise} \end{cases}$$

for a fixed coverage radius r. Or a_{ij} can be derived in other ways that are unrelated to distance, especially in the nonlocation applications of the SCLP discussed below.

The SCLP can be formulated as follows:

$$\text{(SCLP)} \quad \text{minimize} \quad \sum_{j \in J} x_j \tag{8.80}$$

$$\text{subject to} \quad \sum_{j \in J} a_{ij} x_j \ge 1 \qquad \forall i \in I \tag{8.81}$$

$$x_j \in \{0,1\} \qquad \forall j \in J \tag{8.82}$$

The objective function (8.80) calculates the total number of open facilities. Constraints (8.81) ensure that every customer is covered by some open facility (some facility such that both $a_{ij} = 1$ and $x_j = 1$), and constraints (8.82) are integrality constraints.

Sometimes we wish to minimize the total fixed cost of the opened facilities, rather than the total number, in which case the following objective function is appropriate:

$$\text{minimize} \quad \sum_{j \in J} f_j x_j. \tag{8.83}$$

The SCLP is NP-hard (Garey and Johnson 1979). Hakimi (1965) proposed a solution method for the SCLP based on Boolean functions, which has not proven to be effective for realistic-sized instances. Instead, the problem is usually solved using some form of branch-and-bound, an approach first proposed by Toregas et al. (1971). Since the optimal objective function value of the LP relaxation, z_{LP}, is a lower bound on that of (SCLP), and since the optimal objective function value of (SCLP) must be an integer under objective (8.80), Toregas et al. (1971) propose adding the following constraint to (SCLP):

$$\sum_{j \in J} x_j \geq \lceil z_{LP} \rceil. \tag{8.84}$$

Constraint (8.84) acts as a *cut* (see Section 10.3.3), potentially eliminating some fractional solutions without changing the optimal integer solution. The LP relaxation of (SCLP) is usually very tight (and sometimes all-integer) (Bramel and Simchi-Levi 1997), and the addition of constraint (8.84) makes it even tighter.

Toregas and ReVelle (1972) propose row- and column-reduction techniques that can reduce the size of the optimization problem, making it easier to solve. Because of the binary nature of coverage, certain facilities and customers can be eliminated from consideration because they are *dominated*. In particular, a facility j is dominated by a facility k, and we can set $x_j = 0$ (i.e., eliminate column j from the formulation), if $a_{ij} \leq a_{ik}$ for all $i \in I$. In this case, k covers every customer that j serves (and possibly more), so we have no reason to open j. Similarly, a customer i is dominated by a customer ℓ if $a_{ij} \geq a_{\ell j}$ for all $j \in J$. In this case, every facility that covers ℓ also covers i. As long as ℓ is covered by an open facility, so is i, so we can ignore the constraint (row) corresponding to i. See Eiselt and Sandblom (2004) and Daskin (2013) for further discussion of these methods.

☐ **EXAMPLE 8.7**

Return to the 88-node instance from Example 8.4. Using a coverage radius of 400 miles, the optimal solution to the SCLP locates 10 facilities, in El Paso, TX; Fort Worth, TX; Miami, FL; Fresno, CA; Boise City, ID; Tallahassee, FL; Salem, OR; Springfield, IL; Trenton, NJ; and Pierre, SD. See Figure 8.12. Note to achieve total coverage, we needed four more facilities than we needed to obtain 88.7% coverage in Figure 8.11. For example, note the facility in Miami, at the very southern tip of Florida; this facility is in the solution only to cover the city of Miami itself.

☐

8.4.2 The Maximal Covering Location Problem (MCLP)

The SCLP requires every customer to be covered by an open facility. Sometimes this is impractical, because complete coverage would require opening too many facilities. For example, it takes 10 facilities to cover 100% of the demand in the 88-node data set with a 400-mile coverage radius. (See Example 8.6.) But we know from Example 8.7 that we can cover 88.7% of the demand with only six facilities. In fact, if we are only allowed six facilities, we can do better than 88.7%, as we will see below.

The *maximal covering location problem* (MCLP) seeks to maximize the total number of demands covered subject to a limit on the number of open facilities. It was introduced by Church and ReVelle (1974). It uses the same notation as the SCLP, plus the usual parameter p that specifies the allowable number of facilities, as well as a new set of decision variables:

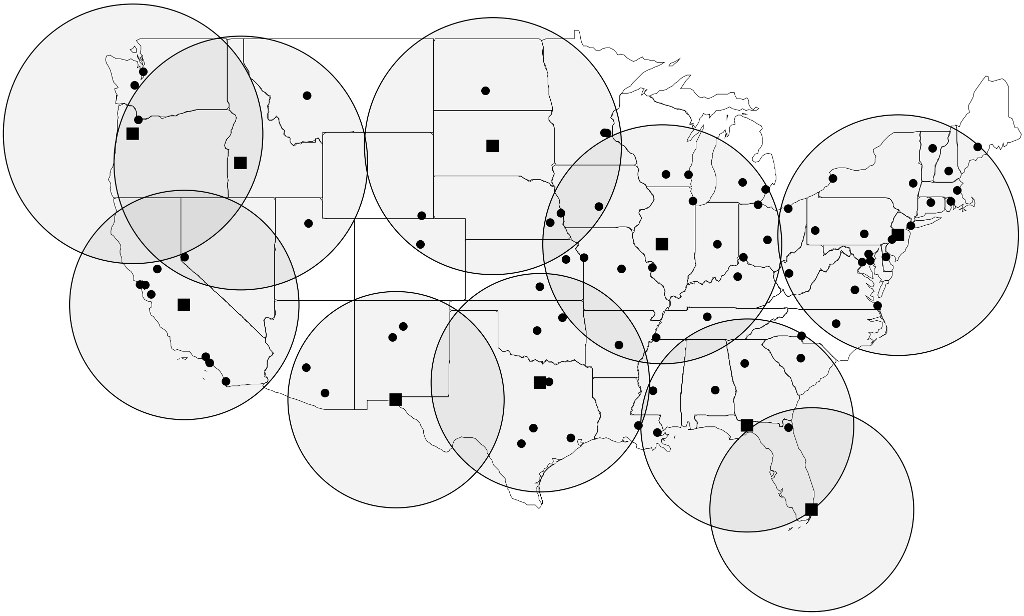


Figure 8.12 Optimal SCLP solution for 88-node instance with coverage radius of 400 miles. 10 facilities are required.

Decision Variables

z_i = 1 if customer $i \in I$ is covered by an open facility, 0 otherwise

The MCLP can be formulated as follows:

$$\text{(MCLP)} \quad \text{maximize} \quad \sum_{i \in I} h_i z_i \tag{8.85}$$

$$\text{subject to} \quad z_i \le \sum_{j \in J} a_{ij} x_j \qquad \forall i \in I \tag{8.86}$$

$$\sum_{j \in J} x_j = p \tag{8.87}$$

$$x_j \in \{0, 1\} \qquad \forall j \in J \tag{8.88}$$

$$z_i \in \{0, 1\} \qquad \forall i \in I \tag{8.89}$$

The objective function (8.85) calculates the total number of covered demands. Constraints (8.86) prevent a customer i from being counted as "covered" unless there is some open facility that covers it. Constraint (8.87) requires exactly p facilities to be opened; as in the pMP, the constraint would be equivalent if we replaced $=$ with $\le$. Constraints (8.88) and (8.89) are integrality constraints. Like the assignment variables y_{ij} in the UFLP, we can relax the z_i variables here to be continuous, and they will always be binary in the optimal solution. (Why?)

The MCLP is NP-hard (Megiddo et al. 1983). Heuristics include a greedy-add heuristic (in which at each iteration, we choose the facility that increases the covered demand the most) and a greedy-add-with-substitution heuristic that considers a "swap" move at each iteration. Both heuristics were proposed by Church and ReVelle (1974). Other heuristics include genetic algorithms (Fazel Zarandi et al. 2011) and another metaheuristic approach called *heuristic concentration* (ReVelle et al. 2008a).

The LP relaxation of (MCLP) tends to be rather tight, and Church and ReVelle (1974) report that 80% of their test instances yielded an all-integer solution for the LP relaxation; Snyder (2011) reports an even higher percentage. Therefore, straightforward LP-based branch-and-bound is often effective. Galvão and ReVelle (1996) propose a Lagrangian relaxation method in which constraints (8.86) are relaxed. The resulting Lagrangian subproblem is:

$$\begin{aligned}
(\text{MCLP-LR}_\lambda) \quad \text{maximize} \quad & \sum_{i \in I} h_i z_i + \sum_{i \in I} \lambda_i \left(\sum_{j \in J} a_{ij} x_j - z_i \right) \\
&= \sum_{i \in I} (h_i - \lambda_i) z_i + \sum_{i \in I} \sum_{j \in J} \lambda_i a_{ij} x_j && (8.90) \\
\text{subject to} \quad & \sum_{j \in J} x_j = p && (8.91) \\
& x_j \in \{0,1\} \qquad \forall j \in J && (8.92) \\
& z_i \in \{0,1\} \qquad \forall i \in I, \forall j \in J && (8.93)
\end{aligned}$$

This subproblem decomposes into two separate problems, one that involves only the x variables and one that involves only the z variables. The x-problem can be solved by setting $x_j = 1$ for the p facilities with the largest values of $\sum_{i \in I} \sum_{j \in J} \lambda_i a_{ij}$. The z-problem can be solved by setting $z_i = 1$ if $h_i - \lambda_i > 0$.

☐ **EXAMPLE 8.8**

Return to the 88-node instance from Example 8.4. Suppose $p = 6$ and we have a coverage radius of 400 miles. Then the optimal solution to the MCLP locates facilities in Fort Worth, TX; Fresno, CA; Madison, WI; Montgomery, AL; Trenton, NJ; and Santa Fe, NM. (See Figure 8.13.) This solution covers 4268 of the 4484 demands, or 95.2%.

Figure 8.14 plots the percentage of demand covered vs. p. From the plot, it is clear that the first several facilities gain a significant percentage of covered demand, whereas subsequent facilities have a diminishing return. When $p \geq 10$, all of the demand is covered, which is what we would expect given that the optimal solution to the SCLP has 10 open facilities (Example 8.7).

☐

8.4.3 The p-Center Problem (pCP)

The third covering problem we discuss is the *p-center problem* (pCP), which minimizes the maximum distance from a customer to its assigned facility while restricting the number of open facilities to p. Although this may not sound at first like a covering problem, the

Figure 8.13 Optimal MCLP solution for 88-node instance with coverage radius of 400 miles and $p = 6$. Total covered demand is 4268 out of 4484 (95.2%).

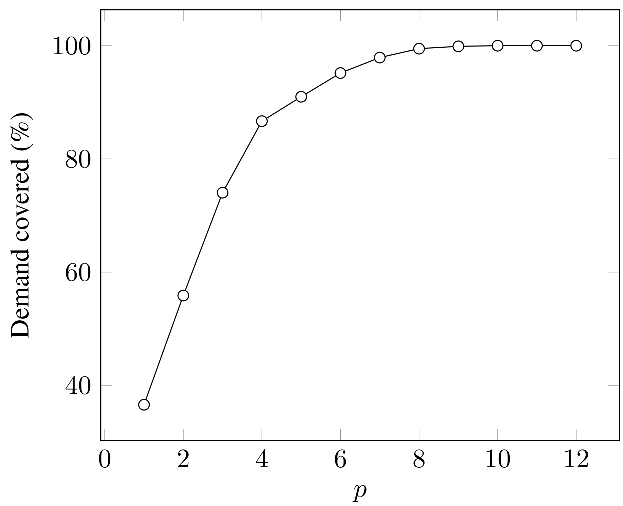


Figure 8.14 Coverage vs. p for 88-node data set with 400-mile coverage radius.

connection can be made explicit by thinking of the pCP as minimizing the coverage radius required to ensure that all customers can be covered by p facilities. Like the SCLP, the pCP aims for an equitable solution, in which no customer is "too far" from an open facility.

For example, in the optimal 6-median solution for the 88-node data set in Figure 8.8, the maximum assigned distance is 801.6 miles, from the customer in Helena, MT, to the facility in Oakland, CA. The pCP asks whether we can make this distance (and all other assigned distances) smaller.

There are two categories of p-center problems: absolute and vertex. In the *absolute p-center problem*, facilities can be located anywhere on the network (i.e., on the vertices or on the links), whereas in the *vertex p-center problem*, facilities can only be located on the vertices of the network. The two are not equivalent since the Hakimi property does not hold for the pCP. (Why?) In this chapter, we consider only vertex p-center problems, and we drop the word "vertex" when referring to the problem. (See Problems 8.37 and 8.38 for algorithms for simple absolute p-center problems.)

The pCP uses notation defined in earlier sections, as well as a single new decision variable:

Decision Variable

r = maximum distance, over all $i \in I$, from i to its assigned facility

In addition, we will tend to think about the parameter c_{ij} as referring to distance, rather than transportation cost, though the distinction is not so important.

The problem can then be formulated as follows:

$$
\begin{array}{llllr}
(p\text{CP}) & \text{minimize} & r & & (8.94) \\
& \text{subject to} & \displaystyle\sum_{j \in J} y_{ij} = 1 & \forall i \in I & (8.95) \\
& & y_{ij} \le x_j & \forall i \in I, \forall j \in J & (8.96) \\
& & \displaystyle\sum_{j \in J} x_j = p & & (8.97) \\
& & \displaystyle\sum_{j \in J} c_{ij} y_{ij} \le r & \forall i \in I & (8.98) \\
& & x_j \in \{0,1\} & \forall j \in J & (8.99) \\
& & y_{ij} \in \{0,1\} & \forall i \in I, \forall j \in J & (8.100)
\end{array}
$$

The objective function (8.94) is simply the maximum assigned distance, r. Constraints (8.95)–(8.97) are identical to (8.70)–(8.72); they require all customers to be assigned, prevent assignments to facilities that are not opened, and require p facilities to be opened. Constraints (8.98) define r by ensuring that it is at least as large as every assigned distance. Constraints (8.99) and (8.100) are integrality constraints on x and y. (The nonnegativity of r is ensured by (8.98).) Note that in the pCP, relaxing (8.100) to $0 \le y_{ij} \le 1$ will *not* ensure integer-valued y variables in the optimal solution. (Why?)

Sometimes we wish to weight the customers differently and minimize the maximum *weighted* assigned distance. In this case, we simply replace (8.98) with

$$\sum_{j \in J} h_i c_{ij} y_{ij} \leq r \qquad \forall i \in I, \tag{8.101}$$

where h_i is the weight on customer $i \in I$.

Like the pMP, the pCP is NP-hard for arbitrary p (Kariv and Hakimi 1979a). Moreover, an off-the-shelf MIP solver such as CPLEX or Gurobi will take orders of magnitude longer to solve (pCP) than any of the other formulations in this chapter. For example, when we solved the 88-node 6-center problem in CPLEX 12.6.1 (see Example 8.9), it took 1607.0 seconds of CPU time. In contrast, it took 0.7 seconds for CPLEX to solve the 6-median problem on the same instance. This is typical of problems like the pCP that have a minimax-type structure, because their LP relaxations tend to be much weaker. For example, the LP relaxation value of the 88-node instance of the pCP is 36.6% smaller than the optimal objective value of the MIP, whereas the LP relaxation of the pMP has an all-integer solution (so the LP and MIP values are equal for this instance).

There is a close relationship between the SCLP and the pCP:

Lemma 8.8 *Let $r \geq 0$. Then the optimal objective function value of the pCP is less than or equal to r if and only if the optimal objective function value of the SCLP with coverage radius r is less than or equal to p.*

Proof. Omitted; see Problem 8.47. ■

This allows us to solve the pCP by exploiting the fact that the SCLP is much easier to solve. In particular, we perform a bisection search on r. For each r, we solve the SCLP. If the optimal objective function value of the SCLP is less than or equal to p, we reduce r, otherwise, we increase it. We continue in this manner until we converge to an r value such that the optimal objective function value of the SCLP equals p but would be larger than p if we made r smaller; this r is the optimal objective function value of the pCP, and the optimal solution to the SCLP is also optimal for the pCP. This approach is typically much faster than solving the MIP (pCP) directly. A method similar to this was first proposed by Minieka (1970).

Algorithm 8.9 summarizes this method in pseudocode. In the algorithm, ϵ is the desired level of optimality tolerance. The inputs r^L and r^U are lower and upper bounds on the optimal r; for example, we can set $r^L = 0$ and $r^U = \max_{i \in I, j \in J}\{c_{ij}\}$. At the end of the algorithm, we use $r = \overline{r}$ since we know for sure that the optimal solution to the SCLP with coverage radius $\overline{r}$ has at most p facilities, but we do not know this for smaller values of r.

□ EXAMPLE 8.9

Let us use Algorithm 8.9 to solve the 6-center problem on the 88-node instance from Example 8.4. We'll set $\epsilon = 0.1$. We begin by setting $r^L = 0$ and $r^U = \max_{i,j}\{c_{ij}\} = 2743.3$. The iterations proceed as follows:

1. $r = 2743.3/2 = 1371.6$; SCLP has optimal objective 2; set $\overline{r} \leftarrow 1371.6$

2. $r = 1371.6/2 = 685.8$; SCLP has optimal objective 4; set $\overline{r} \leftarrow 685.8$

3. $r = 685.8/2 = 342.9$; SCLP has optimal objective 11; set $\underline{r} \leftarrow 342.9$

Algorithm 8.9 SCLP-based algorithm for pCP

input lower and upper bounds r^L and r^U on r
$\underline{r} \leftarrow r^L$; $\overline{r} \leftarrow r^U$ ▷ Initialization
repeat ▷ Main loop
  $r \leftarrow (\underline{r} + \overline{r})/2$ ▷ Candidate value for r
  $x^* \leftarrow$ optimal solution to SCLP with coverage radius r ▷ Solve SCLP
  if $\sum_{j \in J} x_j^* \leq p$ **then** $\overline{r} \leftarrow r$ ▷ Reduce r
  else $\underline{r} \leftarrow r$ ▷ Increase r
  end if
until $\overline{r} - \underline{r} < \epsilon$ ▷ Convergence check
return $x^*, \overline{r}$

4. $r = (342.9 + 685.8)/2 = 514.4$; SCLP has optimal objective 7; set $\underline{r} \leftarrow 514.4$

5. $r = (514.4 + 685.8)/2 = 600.1$; SCLP has optimal objective 6; set $\overline{r} \leftarrow 600.1$

$\vdots$

14. $r = (525.8 + 526.08)/2 = 525.9$; SCLP has optimal objective 7; set $\underline{r} \leftarrow 525.9$

15. $r = (525.9 + 526.08)/2 = 526.0$; SCLP has optimal objective 7; set $\underline{r} \leftarrow 526.0$

At this point, we have $\underline{r} = 526.0$ and $\overline{r} = 526.08$. Since their difference is less than ϵ, the algorithm terminates.

The optimal solution is shown in Figure 8.15. This solution has a maximum assigned distance of 526.06. It locates facilities in Houston, TX; Jacksonville, FL; Tucson, AZ; Omaha, NE; Boise, ID; and Harrisburg, PA.

Using Algorithm 8.9, it took less than 0.5 seconds to find this solution on a laptop computer. In contrast, as noted above, it took over 1600 seconds to solve the MIP (pCP) directly using CPLEX.

□

Most exact algorithms for the pCP proceed along similar lines, though there are some variations. For example, Daskin (2000) proposes an algorithm similar to Algorithm 8.9 but using the MCLP as a subroutine instead of the SCLP. Elloumi et al. (2004) propose a new MIP formulation of the pCP whose LP relaxation is tighter than that of (pCP); they also obtain an even tighter lower bound by relaxing only a subset of the integer variables and show how this bound can be obtained in polynomial time. The bound can then be used in a bisection search similar to that in Algorithm 8.9.

The pCP is polynomially solvable for certain network topologies, such as tree networks (Megiddo et al. 1981, Jeger and Kariv 1985). In some cases, this is true even when c_{ij} is replaced by a nonlinear function of the distance from i to j (Tansel et al. 1982). 1-Center problems on general networks can also be solved in low-order polynomial time, even for absolute pCPs in which the facility may be located at any point along the edges of the network (Kariv and Hakimi 1979a, Shier and Dearing 1983); see also Problems 8.37 and 8.38. Many other results of this type exist; see Tansel (2011) for a review.

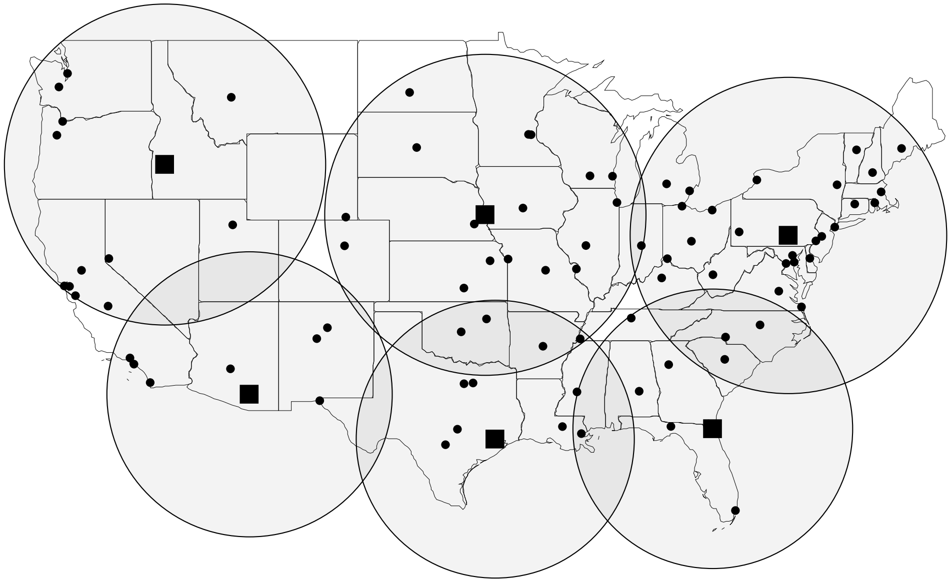


Figure 8.15 Optimal pCP solution for 88-node instance with $p = 6$. Maximum assigned distance = 526.06 miles, from San Jose, CA to facility in Boise, ID.

8.5 OTHER FACILITY LOCATION PROBLEMS

There are many other types of facility location models in the literature and in practice. We mention some other types of location models in this section. For further reading, see the books cited in Section 8.1.

8.5.1 Undesirable Facilities

The problems discussed in this chapter assume that we want customers to be *close* to facilities. In some cases, the opposite is true. For example, when siting toxic waste dumps, weapons repositories, and so on, the goal is usually to locate facilities as far as possible from population centers. In some cases, we want a certain balance: For example, landfills should not be located too close to customers (because of odors, truck traffic, etc.) but also should not be located too far (since garbage collection costs are a function of the distance traveled by the collection trucks).

Problems such as these are known as *undesirable*, *obnoxious*, or *semiobnoxious* facility location problems. One example is the *maxisum location problem*, which seeks to locate p facilities to maximize the sum of the weighted distances between each customer and its nearest open facility. (It is not sufficient to simply change the objective function of the pMP (8.69) from minimize to maximize—why?) See Shamos (1975), Shamos and Hoey (1975), and Church and Garfinkel (1978) for examples of such problems, and see Melachrinoudis (2011) for a review of this literature. Problem 8.31 asks you to formulate the maxisum location problem.

Another type of undesirable location problem is the *p-dispersion problem*, in which we locate p facilities to maximize the minimum distance between any pair of open facilities.

Note that customers are not considered in this model—only facilities. The intent is to ensure that facilities are spread apart as much as possible, as when locating facilities that may interact negatively with one another (such as nuclear power plants) or compete with one another (such as retail locations). See Shier (1977) and Chandrasekaran and Daughety (1981) for early work on this problem. A variant known as the *maxisum dispersion problem* seeks to maximize the sum or average of the distances between pairs of open facilities, rather than the maximum distance. See Kuby (1987) for a discussion of both of these problems.

8.5.2 Competitive Location

Competitive location problems assume that two (or more) firms are locating facilities and that customers will choose a facility to patronize based, at least in part, on distance. These problems are often formulated and analyzed using ideas from game theory (see Section 14.2), in which the goal is to determine a *Nash equilibrium* solution—a solution that neither player wishes to deviate from unilaterally. A Nash equilibrium solution specifies the optimal strategy for both players.

This idea dates back to Hotelling (1929), who considers two competitors who each locate a single facility to serve customers located uniformly along a line (such as a highway or railroad, or, as later authors have suggested, two ice cream vendors on a beach). The firms can locate their facilities anywhere on the line. Hotelling proves that the Nash equilibrium solution is for both players to locate at the midpoint of the line, sharing the demand equally. He also considers how the competitors should set their prices, a factor that has tended to be considered less in subsequent competitive location research. (d'Aspremont et al. (1979) point out a significant error in Hotelling's original work.)

Hotelling's model is a *simultaneous game* in which the two players choose their strategies at the same time, without knowledge of the other's strategy. Most of the more recent work on competitive location has focused on *Stackelberg* or *leader–follower games* in which one player (the leader) moves first, followed by the other player (the follower). Stackelberg games are often modeled as *bilevel* optimization problems in which the optimality of the follower's response is ensured through constraints in the leader's problem. Bilevel problems are difficult in general; see Colson et al. (2007), DeNegre and Ralphs (2009).

Suppose X_p is the set of p facilities that the leader has already located. Then the follower's optimal set of r facilities—the set of facilities that maximizes the follower's captured demand—is called an $(r|X_p)$ *medianoid.* The leader's optimal set X_p of p facilities—the set of facilities that maximizes the leader's captured demand, given that the follower will respond by locating at the $(r|X_p)$ medianoid—is called an $(r|p)$ *centroid.*

Drezner (1982) considers the problem of finding $(r|X_p)$ medianoids and $(r|p)$ centroids on the continuous plane when $r = 1$ and/or $p = 1$. Hakimi (1983) considers medianoids and centroids on networks, showing that the Hakimi property (in which an optimal solution is guaranteed to contain only nodes of the network) does not hold in general and examining medianoids' and centroids' relationships to other problems such as the pMP and pCP. Hakimi proves that the medianoid problem is NP-hard for general r, even when $p = 1$, and that the centroid problem is NP-hard for general p, even when $r = 1$.

ReVelle (1986) focuses on the medianoid (follower's) problem, which he calls the maximum-capture (or MAXCAP) problem. He formulates this problem as an integer programming problem and shows that it is equivalent to the pMP. Serra and ReVelle (1994) consider the centroid (leader's) problem and suggest a heuristic in which the leader locates facilities; the follower solves MAXCAP in response; the leader then updates its facilities using a swap heuristic; then the follower responds by solving MAXCAP; and so on. This type of heuristic, iterating between the leader's and follower's solutions, is common (e.g., Ghosh and Craig (1983)) since the overall bilevel problem is difficult to solve exactly.

For reviews of competitive location models, see Eiselt et al. (1993), Eiselt (2011), Younies and Eiselt (2011), and Dasci (2011).

8.5.3 Hub Location

In some systems, transportation occurs both from facilities to customers and between pairs of facilities. Many airlines use such a structure, offering flights between *hub* airports and from hubs to other cities. To fly between two nonhub cities, one has to fly through one or more hubs and change planes. Similar designs are used in telecommunications and other networks. Such networks are called *hub-and-spoke networks*, and problems that optimize their structure are called *hub location problems*.

A straightforward example of a hub location problem uses the pMP as a starting point. Instead of defining the demand in terms of the nodes (h_i), we define the traffic or flow between nodes i and j as h_{ij}. This traffic must travel from i to a hub k, then to another hub m, and finally to the destination j. It is possible that $k = m$, i.e., the route from i to j travels through only one hub. We wish to

$$\begin{aligned} \text{minimize} \quad & \sum_{j \in J} f_j x_j + \\ & \sum_{i \in I} \sum_{j \in I} h_{ij} \left(\sum_{k \in J} c_{ik} y_{ik} + \alpha \sum_{k \in J} \sum_{m \in J} c_{km} y_{ik} y_{jm} + \sum_{m \in J} c_{jm} y_{jm} \right), \qquad (8.102) \end{aligned}$$

subject to the pMP constraints (8.69)–(8.73) and binary constraints on the y variables. The first term inside the parentheses in (8.102) calculates the cost of i–j traffic as it flows from i to its assigned hub k; the second term is the cost as the i–j traffic travels between hubs k and m; and the third term is the cost of the traffic as it travels from m to j. The second term is discounted by a factor of $\alpha < 1$ to reflect the economies of scale in shipping between hubs. This problem, known as the *p-hub median problem*, was first introduced by O'Kelly (1987).

The primary difficulty with a formulation using (8.102) is that it is nonlinear, due to the second term inside the parentheses. O'Kelly (1987) proposes two enumeration-based heuristics to solve the p-hub median problem. Subsequent papers worked to linearize O'Kelly's formulation. For example, Campbell (1996) introduces binary variables y_{ijkm} that equal the fraction of i–j traffic that is routed through hubs k and m. The resulting formulation is linear but has many more variables ($O(n^4)$ instead of $O(n^2)$, where n is the number of nodes). On the other hand, these variables are continuous rather than binary.

Other hub location problems are based on the UFLP, pCP, and SCLP; see Campbell (1994a) for formulations of these and other problems. For reviews of hub location problems, see Campbell (1994b), Alumur and Kara (2008), and Kara and Taner (2011).

8.5.4 Dynamic Location

During the time when most facilities are operational—years, if not decades—demands and other parameters may change. *Dynamic location problems* model these parameter changes and allow facilities to be added, removed, and/or relocated over time to reflect these changes. Note that we are still assuming that the parameters are deterministic, but that they change over time—they are dynamic.

Ballou (1968) considers the problem of locating and relocating a single facility over a finite planning horizon; he solves the problem heuristically by solving a series of single-period models. Wesolowsky (1973) and Drezner and Wesolowsky (1991) consider a fixed cost for each relocation in the single-facility problem. Scott (1971) considers a multi-facility problem in which one facility is opened per time period; he presents a greedy-type heuristic as well as a dynamic programming approach. Drezner (1995b) generalizes this idea to allow the location of p facilities at any time during T time periods; once open, a facility must remain open. Van Roy and Erlenkotter (1982) consider both openings and closures of facilities over time and solve it using a modified DUALOC algorithm (see Section 8.2.4) embedded in branch-and-bound. Gunawardane (1982) and Schilling (1980) propose dynamic location problems based on coverage objectives.

See Owen and Daskin (1998) for a review of dynamic location problems. Problem 8.51 asks you to formulate a simple example of a dynamic location problem.

8.6 STOCHASTIC AND ROBUST LOCATION MODELS

8.6.1 Introduction

The facility location models we have discussed so far in this chapter are deterministic—they assume that all of the parameters in the model are known with certainty, and that facilities always operate as expected. However, the life span of a typical factory, warehouse, or other facility is measured in years or decades, and over this long time horizon, many aspects of the environment in which the facility operates may change. It is a good idea to anticipate these eventualities when designing the facility network so that the facilities perform well even in the face of uncertainty.

In this section, we discuss approaches for optimizing facility location decisions when the model parameters are stochastic. (In Section 9.6, we discuss a model in which the performance of the facilities itself is stochastic, i.e., the facilities are subject to disruptions.) The stochastic parameters are modeled using *scenarios*, each of which specifies all of the parameters in one possible future state. We must choose facility locations now, before we know which scenario will occur, but we may reassign customers to facilities after we know the scenario. That is, facility locations are *first-stage decisions*, while customer assignments are *second-stage decisions*.

In some models, we know the probability distribution of the scenarios (i.e., the probability that each scenario occurs), while in others we do not. Models in which the probability distribution is known fall under the domain of *stochastic optimization*, while those in which it is not are part of *robust optimization*. In stochastic optimization models, the objective is usually to minimize the expected cost over the scenarios. Several objectives are used for robust facility location models, the most common of which is to minimize the worst-case cost over the scenarios. We will discuss both stochastic and robust approaches for facility location in this section.

Suppose a given set of facilities is meant to operate for 20 years. There are several ways to interpret the way scenarios occur over this time. One way is to assume that we build the facilities today, and then a single scenario occurs tomorrow and lasts for all 20 years. Another is to assume that a new scenario occurs, say, every year or every month, drawn in an iid manner from the scenario distribution. Either interpretation is acceptable for the models we consider in this section.

Choosing the scenarios to include in the model is a difficult task, as much art as science. Expert judgment plays an important role in this process, as can the demand modeling techniques described in Chapter 2. The number of scenarios chosen plays a role in the computational performance of these models: They generally take longer to solve as the number of scenarios increases.

A wide range of approaches for modeling and solving stochastic location problems has been proposed. We discuss only a small subset of them. For more thorough reviews, see Owen and Daskin (1998) or Snyder (2006).

We introduce the following new notation, which we will use throughout this section:

Set

S = set of scenarios

Parameters

h_{is} = annual demand of customer $i \in I$ in scenario $s \in S$

c_{ijs} = cost to transport one unit of demand from facility $j \in J$ to customer $i \in I$ in scenario $s \in S$

q_s = probability that scenario s occurs

Decision Variables

y_{ijs} = the fraction of customer i's demand that is served by facility j in scenario s

Otherwise, the notation is identical to the notation for the UFLP introduced in Section 8.2.2.

8.6.2 The Stochastic Fixed-Charge Location Problem

Suppose we know the scenario probabilities q_s. Our objective is to minimize the total expected cost of locating facilities and then serving customers. We will refer to this problem as the *stochastic fixed-charge location problem* (SFLP). It was formulated by Mirchandani (1980) and Weaver and Church (1983). The SFLP is an example of *stochastic optimization*, a field of optimization that considers optimization under uncertainty. (In particular, this formulation is an example of a *deterministic equivalent* problem.) Usually, the objective is to optimize the expected value of the objective function under all scenarios, and that is the approach we will take here.

The SFLP is formulated as follows:

$$\text{(SFLP)} \quad \text{minimize} \quad \sum_{j \in J} f_j x_j + \sum_{s \in S} \sum_{i \in I} \sum_{j \in J} q_s h_{is} c_{ijs} y_{ijs} \tag{8.103}$$

$$\text{subject to} \quad \sum_{j \in J} y_{ijs} = 1 \qquad \forall i \in I, \forall s \in S \tag{8.104}$$

$$y_{ijs} \le x_j \qquad \forall i \in I, \forall j \in J, \forall s \in S \tag{8.105}$$
$$x_j \in \{0,1\} \qquad \forall j \in J \tag{8.106}$$
$$y_{ijs} \ge 0 \qquad \forall i \in I, \forall j \in J, \forall s \in S \tag{8.107}$$

The objective function (8.103) computes the total fixed plus expected transportation cost. Constraints (8.104) and (8.105) are multiscenario versions of the assignment and linking constraints, respectively. Constraints (8.106) require the location (x) variables to be binary, and constraints (8.107) require the assignment (y) variables to be nonnegative. Note that, if $|S| = 1$, this problem is identical to the classical UFLP. (Therefore, the SFLP is NP-hard.)

The SFLP can be solved using a straightforward modification of the Lagrangian relaxation algorithm for the UFLP (Section 8.2.3). We relax constraints (8.104) to obtain the following Lagrangian subproblem:

(SFLP-LR$_\lambda$)

$$\text{minimize} \quad \sum_{j\in J} f_j x_j + \sum_{i\in I}\sum_{j\in J}\sum_{s\in S} q_s h_{is} c_{ijs} y_{ijs} + \sum_{i\in I}\sum_{s\in S}\lambda_{is}\left(1 - \sum_{j\in J} y_{ijs}\right)$$
$$= \sum_{j\in J} f_j x_j + \sum_{i\in I}\sum_{j\in J}\sum_{s\in S}(q_s h_{is} c_{ijs} - \lambda_{is}) y_{ijs} + \sum_{i\in I}\sum_{s\in S}\lambda_{is} \tag{8.108}$$

$$\text{subject to} \quad y_{ijs} \le x_j \qquad \forall i \in I, \forall j \in J, \forall s \in S \tag{8.109}$$
$$x_j \in \{0,1\} \qquad \forall j \in J \tag{8.110}$$
$$y_{ijs} \ge 0 \qquad \forall i \in I, \forall j \in J, \forall s \in S \tag{8.111}$$

Just as for the UFLP, this problem can be solved easily by inspection. The benefit of opening facility j is

$$\beta_j = \sum_{i\in I}\sum_{s\in S}\min\{0, q_s h_{is} c_{ijs} - \lambda_{is}\}.$$

An optimal solution to (SFLP-LR$_\lambda$) can be found by setting

$$x_j = \begin{cases} 1, & \text{if } \beta_j + f_j < 0 \\ 0, & \text{otherwise} \end{cases}$$
$$y_{ijs} = \begin{cases} 1, & \text{if } x_j = 1 \text{ and } h_{is} c_{ijs} - \lambda_{is} < 0 \\ 0, & \text{otherwise.} \end{cases}$$

The objective value of this solution is given by

$$\sum_{j\in J}\min\{0, \beta_j + f_j\} + \sum_{i\in I}\sum_{s\in S}\lambda_{is}.$$

Upper bounds can be obtained from feasible solutions that are constructed by opening the facilities for which $x_j = 1$ in the Lagrangian subproblem and then assigning each customer to its nearest open facility in each scenario. (Since the transportation cost may vary by scenario, so may the optimal assignments.) The remainder of the Lagrangian relaxation algorithm is similar to that for the UFLP.

The SFLP can actually be interpreted as a special case of the deterministic UFLP obtained by replacing the customer set I with $I \times S$. That is, think of creating multiple instances of each customer, one per scenario, and using this as the customer set. Viewed in that light, the formulation and algorithm for the SFLP are identical to those for the UFLP. This means that an instance of the SFLP with 100 nodes and 10 scenarios is equivalent to an instance of the UFLP with 1000 nodes and can be solved equally quickly.

In fact, the SFLP can also be interpreted another way. Imagine a deterministic problem with multiple products, each of which has its own set of demands and transportation costs. The formulation for SFLP models this situation exactly, so long as we interpret S as the set of products rather than scenarios.

8.6.3 The Minimax Fixed-Charge Location Problem

In this section, we discuss the *minimax fixed-charge location problem* (MFLP), which minimizes the maximum (i.e., worst-case) cost over all scenarios. Minimax problems are an example of *robust optimization*. Robust optimization takes many forms, but the general objective of all of them is to find a solution that performs well no matter how the random variables are realized. Most robust models (including the MFLP) assume that no probabilistic information is known about the random parameters. This is one of the main advantages of robust optimization, since scenario probabilities can be very difficult to estimate. On the other hand, robust optimization problems are generally more difficult to solve than stochastic optimization problems, because of their minimax structure (like the pCP). Moreover, minimax models are often criticized for being overly conservative since their solutions are driven by a single scenario, which may be unlikely to occur. Nevertheless, they are an important class of problems, both within facility location and in robust optimization in general.

Conceptually, the MFLP can be formulated as follows:

$$\text{minimize} \quad \max_{s \in S} \left\{ \sum_{j \in J} f_j x_j + \sum_{i \in I} \sum_{j \in J} h_{is} c_{ijs} y_{ijs} \right\} \tag{8.112}$$

subject to the same constraints as in (SFLP). However, this is not a valid objective function for a linear integer program (because of the "max"), so instead we introduce a new variable, w, that represents the maximum cost over all the scenarios. The MFLP can then be formulated as follows:

$$\text{(MFLP)} \quad \text{minimize} \quad w \tag{8.113}$$

$$\text{subject to} \quad \sum_{j \in J} y_{ijs} = 1 \qquad \forall i \in I, \forall s \in S \tag{8.114}$$

$$y_{ijs} \le x_j \qquad \forall i \in I, \forall j \in J, \forall s \in S \tag{8.115}$$

$$\sum_{j \in J} f_j x_j + \sum_{i \in I} \sum_{j \in J} h_{is} c_{ijs} y_{ijs} \le w \qquad \forall s \in S \tag{8.116}$$

$$x_j \in \{0, 1\} \qquad \forall j \in J \tag{8.117}$$

$$y_{ijs} \ge 0 \qquad \forall i \in I, \forall j \in J, \forall s \in S \tag{8.118}$$

Constraints (8.116) ensure that w is at least as large as the cost in each scenario. Since the objective function (8.113) minimizes w, we are guaranteed that w will *equal* the maximum cost over all scenarios. The remaining constraints are identical to those in (SFLP).

Unfortunately, facility location problems that minimize the worst-case cost, such as the MFLP, are generally much more difficult to solve than their stochastic counterparts. The Lagrangian relaxation algorithm from Section 8.6.2, and most other algorithms for stochastic location problems, cannot be readily adapted for robust problems. Therefore, these problems are generally solved heuristically (e.g., Serra et al. 1996, Serra and Marianov 1998), or solved exactly for special cases such as locating single facilities or locating facilities on specialized networks such as trees (e.g., Vairaktarakis and Kouvelis 1999). Additional results are sometimes possible if the uncertain parameters are modeled using intervals in which the parameters are guaranteed to lie rather than scenarios (e.g., Chen and Lin 1998, Averbakh and Berman 2000a,b).

Another common approach for robust optimization is to minimize the worst-case regret (rather than cost). The *regret* of a given solution in a given scenario is defined as the difference between the cost of that solution in that scenario and the cost of the optimal solution for that scenario. In other words, it's the difference between how well your solution performs in a given scenario and how well you *could have* done if you had known that that scenario would be the one to occur. The *absolute regret* calculates the absolute difference in cost, whereas the *relative regret* reports this difference as a fraction of the optimal cost. If (x, y) is the solution to a facility location problem and $z_s(x, y)$ is the cost of that solution in scenario s, then the absolute regret of (x, y) in scenario s is given by

$$z_s(x, y) - z_s(x_s^*, y_s^*)$$

and the relative regret is given by

$$\frac{z_s(x, y) - z_s(x_s^*, y_s^*)}{z_s(x_s^*, y_s^*)},$$

where (x_s^*, y_s^*) is the optimal solution for scenario s.

Minimax-regret models are closely related to minimax-cost models. In fact, the MFLP can be modified easily to minimize the worst-case regret rather than the worst-case cost simply by subtracting $z_s(x_s^*, y_s^*)$ from the left-hand side of (8.116) (to minimize absolute regret) and by also dividing the left-hand side of (8.116) by $z_s(x_s^*, y_s^*)$ (to minimize relative regret). The constants $z_s(x_s^*, y_s^*)$ must be calculated ahead of time by solving $|S|$ single-scenario problems. Since we are modifying constraints by adding and multiplying constants, the structure of the problem does not change (though the optimal solutions might). Therefore, solutions methods for minimax-cost problems are often applicable for minimax-regret problems, and vice-versa.

8.7 SUPPLY CHAIN NETWORK DESIGN

The facility location models discussed so far in this chapter make decisions about which facilities to open in only a single echelon (the DCs). In practice, firms must often make open/close decisions about multiple echelons (suppliers, factories, etc.), as well as about the transportation links connecting them. We will refer to these more complicated optimization problems as *supply chain network design problems*.

Roughly speaking, supply chain network design problems fall into two categories: *node design* problems, in which we must decide which nodes (facilities) to open, and *arc design* problems, in which we must decide which arcs (links) to open. Both types of problems typically allow for multiple commodities, capacitated nodes and/or arcs, and other side constraints. Facility location problems are examples of relatively simple node design problems.

In some cases, problems of one type can be converted to problems of the other type through suitable modeling tricks such as adding dummy nodes or arcs, and so on. Moreover, some supply chain network design models consider open/close decisions for both nodes and arcs. Nevertheless, we will draw a distinction between the two types of problems and will discuss each type separately: node design problems in Section 8.7.1 and arc design problems in Section 8.7.2.

Although we discuss supply chain network design models in the context of transportation networks, these models are also widely applied in other arenas such as telecommunications, energy, water distribution, and so on.

We will tend to avoid the more generic phrase "network design" since it means different things to different people. To optimizers and other operations researchers, "network design" usually refers to arc design models of the type described in Section 8.7.2, whereas to supply chain practitioners, it usually connotes node design models like those in Section 8.7.1.

8.7.1 Node Design

8.7.1.1 Introduction In this section, we present a model that makes location decisions about two echelons and can be extended to consider a general number of echelons. In addition, this model considers multiple products and joint capacity constraints that reflect the limited capacity in each facility that the several products "compete" for. This problem can be thought of as a multiechelon, multicommodity, capacitated facility location problem. Models such as these are at the core of many commercial supply chain network design software packages.

The seminal paper on multiechelon facility location problems is by Geoffrion and Graves (1974), which presents a three-echelon (plant–DC–customer) model. This paper considers location decisions only at the DC echelon, but it optimizes product flows among all three echelons. The model we will present in this section also considers location decisions at the plant echelon. It is adapted from Pirkul and Jayaraman (1996).

8.7.1.2 Problem Statement This problem is concerned with a three-echelon system consisting of plants, DCs, and customers. The customer locations are fixed, but the plant and DC locations are to be optimized. (See Figure 8.16.) In addition, the model considers multiple products and limited capacity at the plants and DCs. As in the UFLP and CFLP, the objective is to minimize the total fixed and transportation cost.

We will use the following notation:

Sets

I = set of customers
J = set of potential DC locations
K = set of potential plant locations
L = set of products

Demands and Capacities

h_{il} = annual demand of customer $i \in I$ for product $l \in L$
v_j = capacity of DC $j \in J$
b_k = capacity of plant $k \in K$

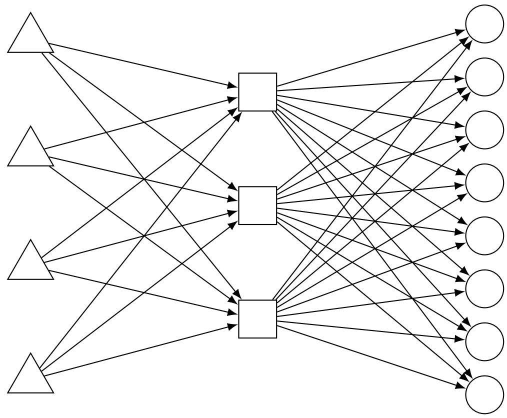


Figure 8.16 Three echelons in node design problem: plants (△), DCs (□), and customers (◯). Customer locations are fixed; plant and DC locations are to be determined by the model.

s_l = units of capacity consumed by one unit of product $l \in L$

Costs

f_j = fixed (annual) cost to open a DC at site $j \in J$
g_k = fixed (annual) cost to open a plant at site $k \in K$
c_{ijl} = cost to transport one unit of product $l \in L$ from DC $j \in J$ to customer $i \in I$
d_{jkl} = cost to transport one unit of product $l \in L$ from plant $k \in K$ to DC $j \in J$

Decision Variables

x_j = 1 if DC j is opened, 0 otherwise
z_k = 1 if plant k is opened, 0 otherwise
y_{ijl} = number of units of product l shipped from DC j to customer i
w_{jkl} = number of units of product l shipped from plant k to DC j

The usage parameter s_l must be expressed in the same units used to express the capacities v_j and c_k. That is, if capacities are expressed in square feet, then s_l is the number of square feet taken up by one unit of product l. If capacities are expressed in person-hours of work available per year, then s_l is the number of person-hours of work required to process one unit of product l. And so on.

The transportation variables y and w indicate the amount of product l shipped along each arc, from plants to DCs (w) and from DCs to customers (y). There is an alternate way to formulate a model like this in which we define a single set of transportation variables, call it y_{ijkl}, that specifies the amount of product l shipped from plant k to customer i via DC j. (Geoffrion and Graves (1974) use this approach.) This type of formulation is more compact and has certain attractive structural properties. However, this strategy requires $|I||J||K||L|$ transportation variables, which is generally larger than the $|I||J||L| + |J||K||L|$ variables required by the formulation below.

Moreover, the strategy of defining a new set of transportation variables for each pair of consecutive echelons allows us to extend this model to more than three echelons. The number of such variables in the alternate approach grows multiplicatively with the number of echelons, while the approach taken here grows only additively.

Note that while in the UFLP, the y_{ij} variables indicated the *fraction* of i's demand served by j, here y_{ijl} is a *quantity*.

8.7.1.3 Formulation The multiechelon location problem can be formulated as a mixed-integer programming (MIP) problem as follows:

$$\text{minimize} \quad \sum_{j\in J} f_j x_j + \sum_{k\in K} g_k z_k + \sum_{l\in L}\left[\sum_{j\in J}\sum_{i\in I} c_{ijl} y_{ijl} + \sum_{k\in K}\sum_{j\in J} d_{jkl} w_{jkl}\right] \tag{8.119}$$

$$\text{subject to} \quad \sum_{j\in J} y_{ijl} = h_{il} \qquad \forall i\in I, \forall l\in L \tag{8.120}$$

$$\sum_{i\in I}\sum_{l\in L} s_l y_{ijl} \le v_j x_j \qquad \forall j\in J \tag{8.121}$$

$$\sum_{k\in K} w_{jkl} = \sum_{i\in I} y_{ijl} \qquad \forall j\in J, \forall l\in L \tag{8.122}$$

$$\sum_{j\in J}\sum_{l\in L} s_l w_{jkl} \le b_k z_k \qquad \forall k\in K \tag{8.123}$$

$$x_j, z_k \in \{0,1\} \qquad \forall j\in J, \forall k\in K \tag{8.124}$$

$$y_{ijl}, w_{jkl} \ge 0 \qquad \forall i\in I, \forall j\in J, \forall k\in K, \forall l\in L \tag{8.125}$$

The objective function (8.119) computes the total fixed and transportation cost. Constraints (8.120) require the total amount of product l shipped to customer i to equal i's demand for l. These constraints are analogous to constraints (8.4) in the UFLP. Constraints (8.121) ensure that the total amount shipped out of DC j is no more than the DC's capacity, and that nothing is shipped out if DC j is not opened. Constraints (8.122) require the total amount of product l shipped into DC j to equal the total amount shipped out. Constraints (8.123) are capacity constraints at the plants and prevent product from being shipped from plant k if k has not been opened. Finally, constraints (8.124) and (8.125) are integrality and nonnegativity constraints.

The UFLP and CFLP are special cases of this problem, and hence it is NP-hard. We will discuss a Lagrangian relaxation algorithm for solving it.

8.7.1.4 Lagrangian Relaxation We will solve the multiechelon location problem using Lagrangian relaxation. Before we do, though, we'll add a new set of constraints to the model:

$$y_{ijl} \le h_{il} \qquad \forall i\in I, \forall j\in J, \forall l\in L \tag{8.126}$$

These constraints simply say that the amount of product l shipped to customer i cannot exceed i's demand for l. They are redundant in the original model in the sense that they are satisfied by every feasible solution. However, they will not be redundant after we relax some of the original constraints. Adding constraints (8.126) tightens the relaxation, as we will see below.

We relax the assignment constraints (8.120) (as in the UFLP) as well as the "balance" constraints (8.122). We use Lagrange multipliers λ_{il} for the first set of constraints and μ_{jl} for the second. The resulting subproblem is as follows:

$$\text{minimize} \quad \sum_{j \in J} f_j x_j + \sum_{k \in K} g_k z_k + \sum_{l \in L} \left[\sum_{j \in J} \sum_{i \in I} c_{ijl} y_{ijl} + \sum_{k \in K} \sum_{j \in J} d_{jkl} w_{jkl} \right]$$

$$+ \sum_{i \in I} \sum_{l \in L} \lambda_{il} \left(h_{il} - \sum_{j \in J} y_{ijl} \right) + \sum_{j \in J} \sum_{l \in L} \mu_{jl} \left(\sum_{i \in I} y_{ijl} - \sum_{k \in K} w_{jkl} \right) \quad (8.127)$$

$$\text{subject to} \quad y_{ijl} \le h_{il} \quad \forall i \in I, \forall j \in J, \forall l \in L \quad (8.128)$$

$$\sum_{i \in I} \sum_{l \in L} s_l y_{ijl} \le v_j x_j \quad \forall j \in J \quad (8.129)$$

$$\sum_{j \in J} \sum_{l \in L} s_l w_{jkl} \le b_k z_k \quad \forall k \in K \quad (8.130)$$

$$x_j, z_k \in \{0, 1\} \quad \forall j \in J, \forall k \in K \quad (8.131)$$

$$y_{ijl}, w_{jkl} \ge 0 \quad \forall i \in I, \forall j \in J, \forall k \in K, \forall l \in L \quad (8.132)$$

The first two sets of constraints involve only the x and y variables, while the third set involves only the z and w variables. This allows us to decompose the subproblem into two separate subproblems:

$$(xy\text{-problem}) \quad \text{minimize} \quad \sum_{j \in J} f_j x_j + \sum_{i \in I} \sum_{j \in J} \sum_{l \in L} (c_{ijl} - \lambda_{il} + \mu_{jl}) y_{ijl} \quad (8.133)$$

$$\text{subject to} \quad y_{ijl} \le h_{il} \quad \forall i \in I, \forall j \in J, \forall l \in L \quad (8.134)$$

$$\sum_{i \in I} \sum_{l \in L} s_l y_{ijl} \le v_j x_j \quad \forall j \in J \quad (8.135)$$

$$x_j \in \{0, 1\} \quad \forall j \in J \quad (8.136)$$

$$y_{ijl} \ge 0 \quad \forall i \in I, \forall j \in J, \forall l \in L \quad (8.137)$$

$$(zw\text{-problem}) \quad \text{minimize} \quad \sum_{k \in K} g_k z_k + \sum_{k \in K} \sum_{j \in J} \sum_{l \in L} (d_{jkl} - \mu_{jl}) w_{jkl} \quad (8.138)$$

$$\text{subject to} \quad \sum_{j \in J} \sum_{l \in L} s_l w_{jkl} \le b_k z_k \quad \forall k \in K \quad (8.139)$$

$$z_k \in \{0, 1\} \quad \forall k \in K \quad (8.140)$$

$$w_{jkl} \ge 0 \quad \forall j \in J, \forall k \in K, \forall l \in L \quad (8.141)$$

Both problems are quite easy to solve. First, consider the xy-problem. If we set $x_j = 1$ for a given j, then we are allowed to set some of the y_{ijl} variables to something greater than 0. The problem of determining values for the y_{ijl} variables (assuming $x_j = 1$) is a continuous knapsack problem. Here's where constraints (8.126) come into play. If we didn't have these constraints in the formulation, we would set $y_{ijl} = v_j / s_l$ for only a single i and l. By imposing bounds on the y_{ijl} variables, we obtain a solution that is much closer to the true optimal solution and hence provides a tighter lower bound. For each j, we solve the continuous knapsack problem, and if the optimal objective value is less than $-f_j$, we set $x_j = 1$; otherwise, we set $x_j = 0$. Solving the zw-problem is very similar, except that there are no explicit upper bounds on the w_{jkl} variables.

As in the UFLP, upper bounds are found using a greedy-type heuristic, and the Lagrange multipliers are updated using subgradient optimization. In computational tests reported by Pirkul and Jayaraman (1996), this algorithm could solve small- to medium-sized problems in roughly 1 minute.

☐ **EXAMPLE 8.10**

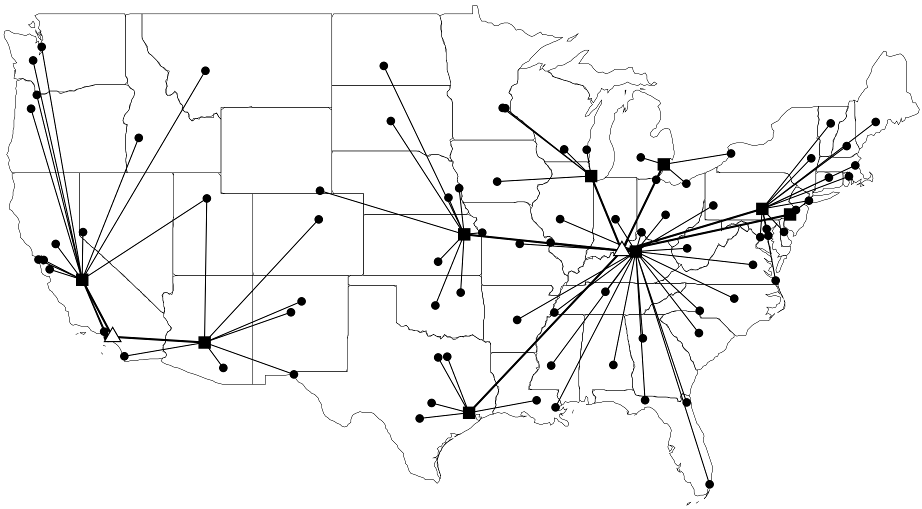
Let us broaden the scope of the 88-node UFLP instance to include plant-location decisions, as well as multiple products. The data set `node_design_data.xlsx` considers a set K of 10 possible locations for plants, each of which has a total capacity of 10,000 units and a fixed cost of $\$1{,}000{,}000$. The set L of products in this data set consists of five products, of which product 1 is from the original 88-node data set. We assume that each DC in J has a capacity of 2000 units and the same fixed costs as in the UFLP instance. Transportation costs d_{jkl} are set equal to 0.25 times the great circle distance between plant k and DC j, whereas c_{ijl} continues to equal 0.5 times the distance between DC j and customer i. (Plant–DC shipments are typically larger and therefore benefit from economies of scale; hence the smaller per-unit costs.) Transportation costs are the same for every product.

The optimal solution to this 98-node instance of the node design problem is shown in Figure 8.17(a). This solution opens nine DCs (in Chicago, IL; Houston, TX; Philadelphia, PA; Detroit, MI; Phoenix, AZ; Fresno, CA; Topeka, KS; Harrisburg, PA; and Frankfort, NY) and two plants (in Louisville, KY, and Anaheim, CA). The plants are drawn as triangles in Figure 8.17(a). To make the plant–DC shipments easier to visualize, Figure 8.17(b) draws the customers and their inbound links in a lighter shade.

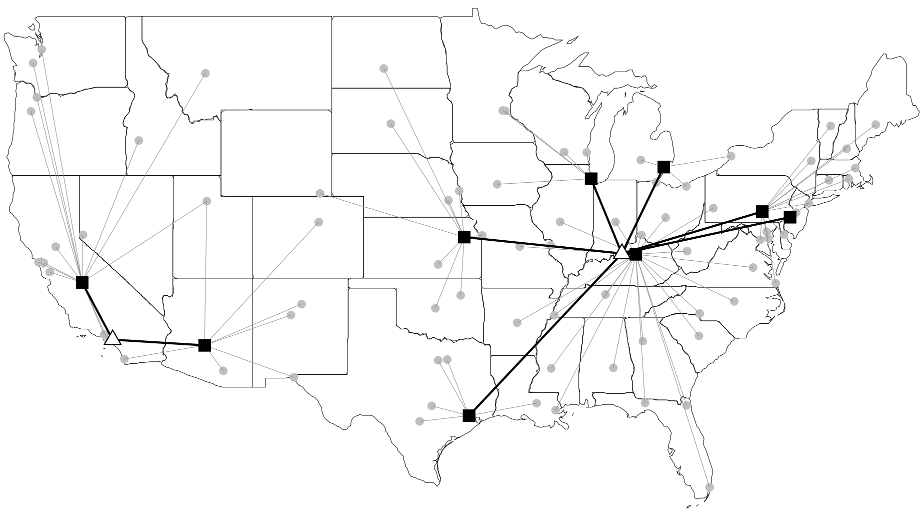
In the optimal solution, two customers are served from more than one DC. In particular, the customer in New York, NY, receives product 1 from both Philadelphia, PA, and Harrisburg, PA; it receives all other products only from Philadelphia. In addition, the customer in Salt Lake City, UT, receives products 1 and 2 from Phoenix, AZ, and the other products from Fresno, CA. At first, it may seem surprising that only 2 of the 88 customers are served by multiple DCs, but this is actually fairly typical; when a given facility's capacity is fully utilized, only the "final" customer will have its demand split. In this solution, only two of the DCs (Fresno and Philadelphia) are fully utilized.

This solution is different from the solution we would obtain by following a sequential approach in which we first solve for the optimal DC locations (ignoring the plants), then fix open the resulting DCs and find the optimal plant locations. That solution, which is pictured in Figure 8.18, opens 11 DCs and the same 2 plants as the optimal solution. It has a total cost of \$4,776,380, which is only 2.1% more expensive than optimal. In general, however, the error from this sequential approach can be considerably larger.

☐

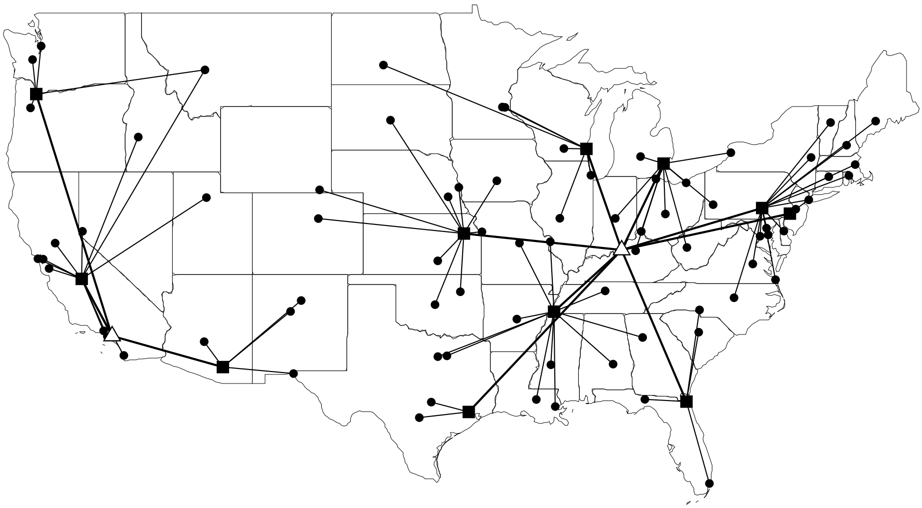


(a) All shipments.

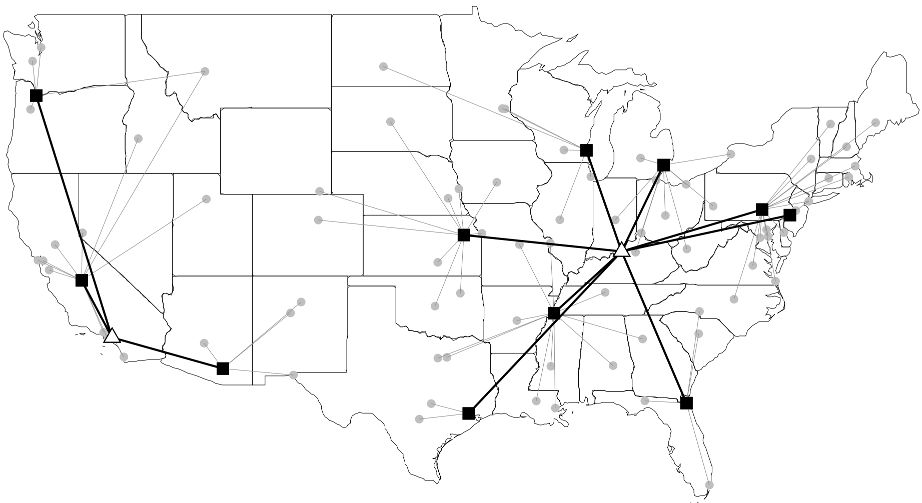


(b) Plant–DC shipments.

Figure 8.17 Optimal solution to 98-node node design instance. Total cost = $4,678,145.



(a) All shipments.



(b) Plant–DC shipments.

Figure 8.18 Sequential-optimization solution to 98-node node design instance. Total cost = $4,776,380.

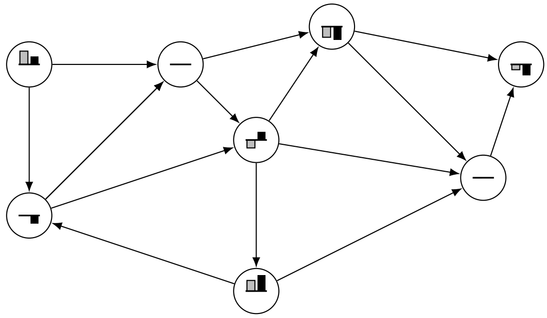


Figure 8.19 Simple arc design problem instance. Grey and black bars inside nodes indicate supply (> 0) or demand (< 0) for two products.

8.7.2 Arc Design

We now turn our attention to arc design problems, in which the nodes of the network are already determined, and we are to make decisions about which arcs (or links, or edges) to open. These are classical OR models; indeed, as noted above, the generic phrase "network design" often connotes this type of problem when used by an optimizer or other operations researcher. For a more thorough discussion of arc design problems, see Magnanti and Wong (1984).

8.7.2.1 Problem Statement We are given a set N of nodes (which are already open) and a set E of potential arcs. We will assume that the arcs are *directed*, i.e., that arc (i, j) is not the same as arc (j, i) for $i, j \in N$. To model undirected networks, we can simply double each arc, orienting one copy in each direction.

The network can handle multiple products (commodities), which are contained in the set L. Each node $i \in N$ has a certain number b_i^l of "available units" of product $l \in L$: If $b_i^l > 0$, then node i supplies b_i^l units of product l to the network; if $b_i^l < 0$, then node i demands $-b_i^l$ units of product l; and if $b_i^l = 0$, then node i does neither. In any of these cases, product l may flow through node i en route to other nodes.

We will assume that if node i supplies product l ($b_i^l > 0$), then it must send *exactly* b_i^l units into the network. This implies that the total supply of product l equals the total demand:

$$\sum_{i \in N} b_i^l = 0$$

for all $l \in L$. It is simple to relax this assumption by adding a dummy node that "absorbs" the excess supply if $\sum_{i \in N} b_i^l > 0$.

For example, Figure 8.19 depicts a simple instance for an arc design problem with two products. The small bar graph inside each node indicates the available units of each of the two products. For example, the node in the top left supplies 5 units of product 1 and 3 units of product 2; the node in the middle demands 3 units of product 1 and supplies 3 units of product 2; the node in the bottom right does not supply or demand any units of either product; and so on.

If we open arc (i, j), we incur a fixed cost of f_{ij}. If it is opened, we can send product flows along arc (i, j), at a cost of c_{ij}^l per unit of product l. Arc (i, j) has a total capacity of v_{ij} units of flow (summed across all products). We assume that the capacity v_{ij} is expressed in the same units as the b_i^l values. We use decision variables x_{ij} and y_{ij}^l to denote whether arc (i, j) is opened and the flow of product l on arc (i, j), respectively.

In addition to the constraints described above, we may include other "side constraints." (See Problem 8.57 for some examples.) We let S denote the set of solutions (x, y) that are feasible with respect to these side constraints. If there are no side constraints, we can set S equal to a set that does not impose any additional constraints, such as

$$S = \{0, 1\}^{|E|} \times \mathbb{R}_+^{|E||L|}, \tag{8.142}$$

where $\mathbb{R}_+$ is the set of all nonnegative real numbers.

As in the UFLP, the key trade-off in this problem is between the fixed cost to construct arcs and the variable cost in using them. The more arcs we open, the higher our fixed costs, but the more flexibility we have in transporting the products, and therefore, the lower the flow costs.

We summarize the notation below:

Sets

N = set of nodes
E = set of potential arcs
L = set of products
S = set of solutions (x, y) that are feasible with respect to side constraints

Parameters

b_i^l = available units of product $l \in L$ at node $i \in I$
v_{ij} = capacity of arc $(i, j) \in E$
f_{ij} = fixed cost to open arc $(i, j) \in E$
c_{ij}^l = cost to transport one unit of product $l \in L$ along arc $(i, j) \in E$

Decision Variables

x_{ij} = 1 if arc $(i, j) \in E$ is opened, 0 otherwise
y_{ij}^l = number of units of product $l \in L$ shipped along arc $(i, j) \in E$

8.7.2.2 *Formulation*

The arc design problem can be formulated as follows:

$$\text{minimize} \quad \sum_{(i,j)\in E} f_{ij}x_{ij} + \sum_{(i,j)\in E} \sum_{l\in L} c_{ij}^l y_{ij}^l \tag{8.143}$$

$$\text{subject to} \quad \sum_{j\in N} y_{ij}^l - \sum_{j\in N} y_{ji}^l = b_i^l \qquad \forall i \in N, \forall l \in L \tag{8.144}$$

$$\sum_{l\in L} y_{ij}^l \le v_{ij}x_{ij} \qquad \forall (i, j) \in E \tag{8.145}$$

$$(x, y) \in S \tag{8.146}$$

$$x_{ij} \in \{0, 1\} \qquad \forall (i, j) \in E \tag{8.147}$$

$$y_{ij}^l \ge 0 \qquad \forall (i, j) \in E, \forall l \in L \tag{8.148}$$

The objective function calculates the total fixed cost plus flow costs over all arcs and products. Constraints (8.144) are flow balance constraints: They require the net flow out of node i of product l (flow out minus flow in) to equal the available supply of l at node i. If $b_i^l > 0$, then more units of l flow out of than into node i; if $b_i^l < 0$, then more units flow in than out; and if $b_i^l = 0$, then all units that enter node i also leave it. Constraints (8.145) prevent flow along an arc that has not been opened, and also enforce the capacity (joint across all products) on the arc. Constraints (8.146) are the side constraints. Constraints (8.147) and (8.148) are integrality and nonnegativity constraints.

This problem is NP-hard; the easiest way to see this is to note that many well known NP-hard problems are special cases of it. Even if the x variables are fixed, the problem is still difficult to solve: It becomes a multicommodity network flow problem. If fractional flows are allowed, the multicommodity network flow problem is usually formulated as an LP, but it is a large and particularly challenging LP to solve. If the flows must be integer, even finding a feasible solution is NP-complete. (See Ahuja et al. (1993).)

If there are no capacities ($v_{ij} = \infty$ for all $(i,j) \in E$) and no side constraints, the problem is sometimes called the *fixed-charge design problem*. If there are no capacities, no fixed costs ($f_{ij} = 0$ for all $(i,j) \in E$), and a single side constraint consisting of a budget constraint, it is known as the *budget design problem*. Both of these problems are considerably easier to solve than their capacitated counterparts, primarily because the presence of capacities weakens the LP relaxation.

Arc design problems exist in many other flavors. For example, sometimes each product l is assumed to have a *single* origin and destination (Magnanti and Wong 1984); this is common in telecommunications networks in which the flow represents packets that must be routed from one node to another. Sometimes the commodity is even required to follow a single path through the network, rather than being split and recombined (Gavish and Altinkemer 1990). In other models, we must decide how many facilities to open on each arc or, similarly, how much capacity to add to each arc (Bienstock and Günlük 1996, Gendron et al. 1999). Other models include nonlinear costs, arc congestion, dynamically changing parameters, and so on.

8.7.2.3 Solution Methods Uncapacitated arc design problems are frequently solved using Benders decomposition. The basic idea is to choose values for the x variables in a "master problem," solve for the optimal resulting flows in a "subproblem," and then use those flows to determine additional cuts that can be added to the master problem to eliminate the current (infeasible or suboptimal) x and find a better one. For further discussion of Benders decomposition applied to arc design problems, see Magnanti and Wong (1984), Magnanti et al. (1986), and Costa (2005). A dual-ascent procedure based on the DUALOC algorithm was proposed by Balakrishnan et al. (1989).

As noted above, capacitated arc design problems are considerably more difficult to solve. Algorithms have been proposed using branch-and-cut (Günlük 1999, Atamtürk 2002) and Lagrangian relaxation (Holmberg and Yuan 2000, Crainic et al. 2001), among others. For a survey, see Gendron et al. (1999). Heuristics such as add/drop-type methods have been applied to arc design problems (Powell 1986), as have tabu search (Crainic et al. 2000, Ghamlouche et al. 2003), genetic algorithms (Drezner and Salhi 2002), and other metaheuristics.

☐ **EXAMPLE 8.11**

Figure 8.20 maps the ten largest cities in Hungary along with potential arcs connecting them. This instance consists of three products, all of whose demand occurs in Budapest. Product 1 is produced in Szeged and Székesfehérvár; product 2 is produced in Szeged and Debrecen; and product 3 is produced in Pécs and Miskolc. Available units b_i^l are plotted as bar graphs, as in Figure 8.19. The arc widths are proportional

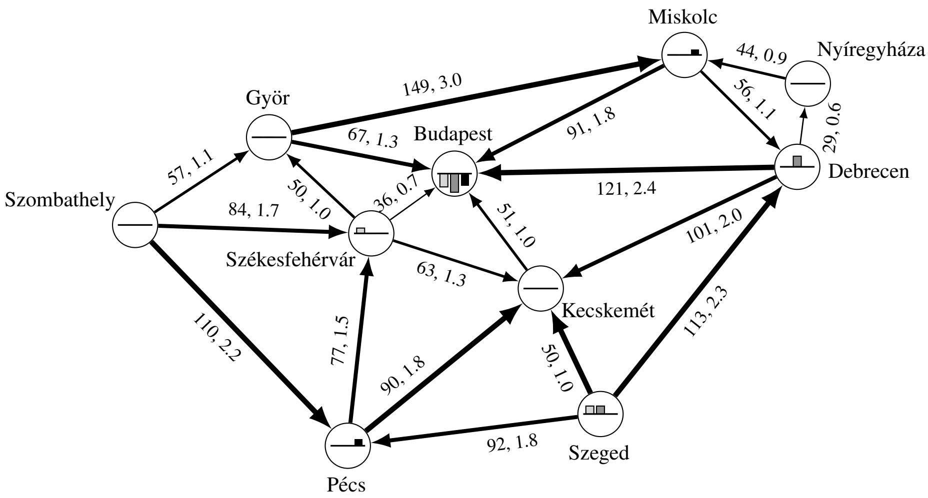


Figure 8.20 Hungary cities arc design problem instance. Grey and black bars inside nodes indicate supply (> 0) or demand (< 0) for three products.

to their capacities, and the fixed and variable costs are listed along the arcs. Variable costs are the same for all products. (For the complete specification of the instance, see the file `hungary.xlsx`.)

The optimal solution to this instance of the arc design problem is drawn in Figure 8.21(a). This solution has a total cost of 647.6. The solution opens 8 arcs. CPLEX solved this instance in less than 1 second.

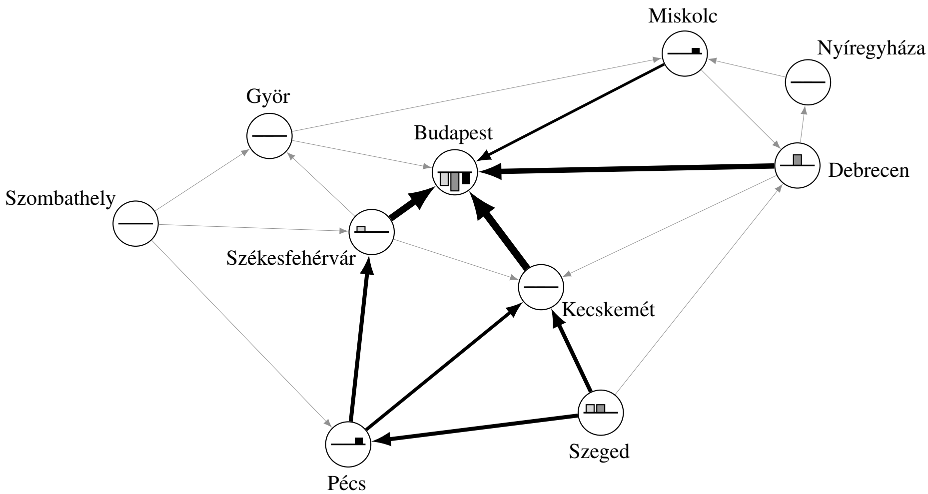
The optimal solution is different from the solution that would result from solving 3 separate single-product problems, one for each product, and then combining the results. That solution is plotted in Figure 8.21(b). In fact, that solution is not even feasible for the original problem, since it sends 6 units of flow along the arc from Szeged to Kecskemét, but that arc only has a capacity of 3. Solving for each product individually ignores the shared capacity and results in this infeasible solution. This highlights the fact that it can be difficult even to find a feasible solution for the capacitated arc design problem, let alone an optimal one.

□

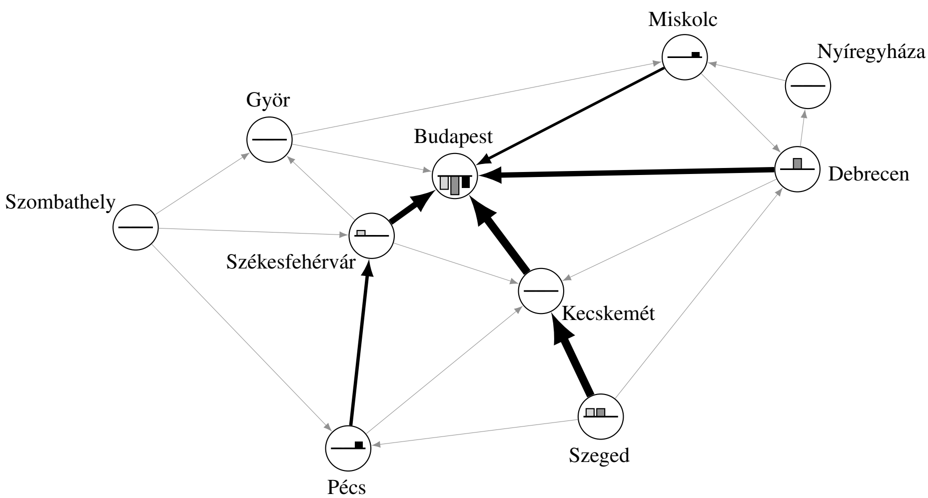
CASE STUDY 8.1 Locating Fire Stations in Istanbul

Istanbul, Turkey, is one of the world's largest cities, with a population of over 13 million and growing. By 2008, the population growth had rendered the existing set of fire stations insufficient to meet the current needs, prompting the Istanbul Metropolitan Municipality (IMM) to sponsor a project by researchers from Dogus University and Istanbul Technical University to determine locations for new fire stations in the city. Their project is described by Aktaş et al. (2013) and summarized here.

The city of Istanbul is divided into 40 districts and 790 subdistricts. The researchers treated these subdistricts as both the demand nodes and potential facility locations in their location models. The IMM aims to respond to fire incidents within 5 minutes,



(a) Optimal solution.



(b) Solution obtained by solving for each product individually.

Figure 8.21 Solutions to Hungary cities arc design problem instance. Arcs selected in the solution are black; nonselected potential arcs are gray.

leading the researchers to use coverage models with a coverage radius based on a five-minute travel time. The researchers used the set covering location problem (SCLP) to identify the cheapest solution that would achieve 100% service, as well as the maximal covering location problem (MCLP) to find solutions that maximize coverage subject to a budget constraint. Both models required all existing fire stations to remain open; the goal was to choose locations for new fire stations.

Their SCLP model differs from the formulation given in Section 8.4.1 in two ways. First, it allows for multiple types of fire stations, each with its own fixed cost and capacity. Second, it requires a given subdistrict to be covered by sufficiently many, or sufficiently large, fire stations to meet the annual number of fire incidents in that subdistrict. In particular, it replaces constraints (8.81) with

$$\sum_{j\in J}\sum_{k\in K} r_k a_{ij} x_{jk} \ge h_i \qquad \forall i \in I, \tag{8.149}$$

where K is the set of fire station types, r_k is the capacity of a type-k station (number of incidents it can handle per year), h_i is the number of incidents in subdistrict i per year, and $x_{jk} = 1$ if we open a fire station of type k in subdistrict j. The model also imposes a constraint requiring at most one type of station to be opened in each subdistrict:

$$\sum_{k\in K} x_{jk} \le 1 \qquad \forall j \in J. \tag{8.150}$$

Finally, the objective function (8.83) is modified to sum over k in addition to j.

Their MCLP model is similarly modified, replacing constraints (8.86) with

$$h_i z_i \le \sum_{j\in J}\sum_{k\in K} r_k a_{ij} x_{jk} \qquad \forall i \in I. \tag{8.151}$$

In other words, i only counts as covered if the opened facilities that cover i have sufficient combined capacity to respond to the number of incidents at i. In some versions of their model, they also modify h_i to reflect the number of cultural heritage sites in the subdistrict and to give more weight to those subdistricts that have more such sites. (The city's history goes back more than 2500 years. A group of sites called the Historical Areas of Istanbul was placed on the UNESCO World Heritage List in 1985.)

The research team used a commercial geographic information system (GIS) to assemble the data for the study. The GIS calculated the geographical center of each subdistrict and the average travel times between subdistricts, taking into account the road network and the typical speed on each road link. These travel times were then used to determine the coverage parameters a_{ij} for the SCLP and MCLP. Fixed location costs were assumed to be the same at every location, but different for different fire station types. Demands h_i were estimated from 12 years of historical incident data from IMM.

The status quo solution, consisting of Istanbul's existing 60 fire stations, was shown to cover only 56.6% of the demands in the model (as measured by historical incidents) within a 5-minute service time, and only 18.2% of demands from subdistricts that contain cultural heritage sites. This poor coverage was the result of the city's expansion or changes in the road system and is what prompted this study in the first place.

The SCLP solution, which covers 100% of all demands, required 149 new stations. This exceeded the IMM's budget for opening new stations, which allowed for the equivalent of 64 new stations. Therefore, the researchers imposed this budget constraint in the MCLP and found a solution that covers 93.9% of the demand, including 71.1% of demands from heritage subdistricts. It also double-covers 35.6% of the subdistricts, more than twice the number that are double-covered in the status quo solution. The problems were solved in the modeling language GAMS using the MIP solver CPLEX, with run times of less than 1 second.

As of their 2013 paper, Aktaş, et al. report that IMM had opened 25 new fire stations in subdistricts proposed by the model, with a subsequent slowdown due to economic conditions. Their solution provides a roadmap for future expansion of the fire station network that can be implemented as budgets allow.

PROBLEMS

8.1 (Locating DCs for Toy Stores) A toy store chain operates 100 retail stores throughout the United States. The company currently ships all products from a central distribution center (DC) to the stores, but it is considering closing the central DC and instead operating multiple regional DCs that serve the retail stores. It will use the UFLP to determine where to locate DCs. Planners at the company have identified 24 potential cities in which regional DCs may be located. The file `toy-stores.xlsx` lists the longitude and latitude for all of the locations (stores and DCs), as well as the annual demand (measured in pallets) at each store and the fixed annual location cost at each potential DC location. Using optimization software of your choice, implement the UFLP model from Section 8.2.2 and solve it using the data provided. Assume that transportation from DCs to stores costs \$1 per mile, as measured by the great circle distance between the two locations. Report the optimal cities to locate DCs in and the optimal total annual cost.

8.2 (10-Node UFLP Instance: Exact) The file `10node.xlsx` contains data for a 10-node instance of the UFLP, with nodes located on the unit square and $I = J$, pictured in Figure 8.22. The file lists the x- and y-coordinates, demands h_i, and fixed costs f_j for each node, as well as the transportation cost c_{ij} between each pair of nodes i and j. Transportation costs equal 10 times the Euclidean distance between the nodes. All fixed costs equal 200.

Solve this instance of the UFLP exactly by implementing the UFLP in the modeling language of your choice and solving it with a MIP solver. Report the optimal locations, optimal assignments, and optimal cost.

8.3 (10-Node UFLP Instance: Greedy-Add) Use the greedy-add heuristic to solve the 10-node UFLP instance described in Problem 8.2. Report the facility that is opened at each iteration, as well as the final locations, assignments, and cost.

8.4 (10-Node UFLP Instance: Swap) Suppose we have a solution to the 10-node UFLP instance described in Problem 8.2 in which $x_2 = x_3 = 1$ and $x_j = 0$ for all other j. Use the swap heuristic to improve this solution. Use a best-improving strategy (that is, search through the facilities in order of index, and at each iteration, implement the first swap

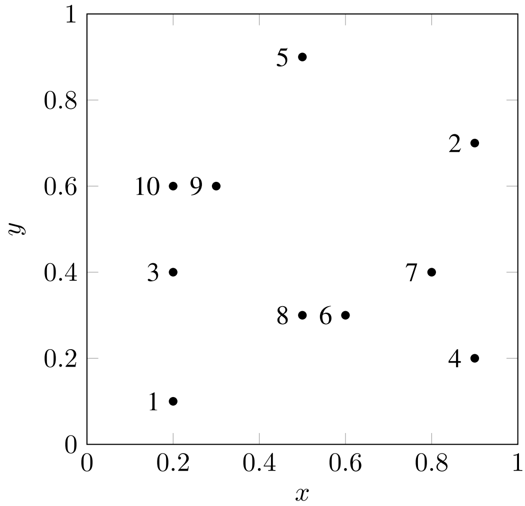


Figure 8.22 10-node facility location instance for Problems 8.2–8.11.

found that improves the cost.) Report the swaps made at each iteration, as well as the final locations, assignments, and cost.

8.5 (10-node pMP Instance: Exact) Using the file `10node.xlsx` (see Problem 8.2), solve the pMP exactly by implementing it in the modeling language of your choice and solving it with a MIP solver. Ignore the fixed costs in the data set and use $p = 4$. Report the optimal locations, optimal assignments, and optimal cost.

8.6 (10-Node pMP Instance: Swap) Suppose we have a solution to the 10-node pMP instance described in Problem 8.5 in which $x_2 = x_3 = x_5 = x_8 = 1$ and $x_j = 0$ for all other j. Use the swap heuristic to improve this solution. Use a best-improving strategy (that is, search through the facilities in order of index, and at each iteration implement the first swap found that improves the cost.) Report the swaps made at each iteration, as well as the final locations, assignments, and cost.

8.7 (10-Node pMP Instance: Neighborhood Search) Suppose we have a solution to the 10-node pMP instance described in Problem 8.5 in which $x_4 = x_5 = x_6 = x_{10} = 1$ and $x_j = 0$ for all other j. Use the neighborhood search heuristic to improve this solution. Report the swaps made at each iteration, as well as the final locations, assignments, and cost.

8.8 (10-node SCLP Instance) Using the file `10node.xlsx` (see Problem 8.2), solve the SCLP exactly by implementing it in the modeling language of your choice and solving it with a MIP solver. Set the fixed cost of every facility equal to 1. Assume that facility j covers customer i if $c_{ij} \leq 2.5$. Report the optimal locations.

8.9 (10-node MCLP Instance) Using the file `10node.xlsx` (see Problem 8.2), solve the MCLP exactly by implementing it in the modeling language of your choice and solving it with a MIP solver. Set $p = 4$. Assume that facility j covers customer i if $c_{ij} \leq 2.5$. Report the optimal locations and the total number of demands covered.

8.10 (10-node MCLP Instance: Coverage vs. p) Using the file `10node.xlsx` (see Problem 8.2), solve the MCLP exactly for $p = 1, 2, \ldots, 10$ using the modeling language and solver of your choice. Assume that facility j covers customer i if $c_{ij} \le 2.5$. Construct a plot similar to Figure 8.14.

8.11 (10-node pCP Instance) Use Algorithm 8.9 to solve the 10-node instance of the pCP specified in the file `10node.xlsx` (see Problem 8.2). Set $p = 3$. Use $r^L = 0$, $r^U = \max_{i \in I, j \in J}\{c_{ij}\}$, and $\epsilon = 0.1$. Report the value of r at each iteration, as well as the optimal locations, assignments, and objective function value.

8.12 (Locating Homework Centers for Chicago Schools) Suppose the City of Chicago wishes to establish homework-help centers at 12 of its public libraries. It wants the homework center locations to be as close as possible to Chicago public schools. In particular, it wants the homework centers to cover as many schools as possible, where a school is "covered" if there is a homework center located within 2 miles of it.

a) Using the files `chicago-schools.csv` and `chicago-libraries.csv` and determining coverage using great circle distances, find the 12 libraries at which homework centers should be established. Report the indices of the libraries selected, as well as the total number of schools covered. (Chicago school and library data are adapted from Chicago Data Portal (2017a,b).)

b) Suppose now that the city wishes to ensure that *all* schools are covered. What is the minimum number of homework centers that must be established to accomplish this?

8.13 (Easy or Hard Modifications?) Which of the following costs can be implemented in the UFLP by modifying the parameters only, without requiring structural changes to the model; that is, without requiring modifications to the variables, objective function, or constraints? Explain your answers briefly.

a) A per-unit cost to ship items from a supplier to facility j. (The cost may be different for each j.)

b) A per-unit processing cost at facility j. (The cost may be different for each j.)

c) A fixed cost to ship items from facility j to customer i. (The cost is independent of the quantity shipped but may be different for each i and j.)

d) A transportation cost from facility j to customer i that is a nonlinear function of the quantity shipped (for example, one of the quantity discount structures discussed in Section 3.4).

e) A fixed capacity-expansion cost that is incurred if the demand served by facility j exceeds a certain threshold.

f) Some facilities are already open; an open facility j can be closed at a cost of $\hat{f}_j$. (In addition, we can open new facilities, as in the UFLP.)

8.14 (LP Relaxation of UFLP) Develop a simple instance of the UFLP for which the optimal solution to the LP relaxation has fractional values of the x_j variables. This solution must be strictly optimal—that is, you can't submit an instance for which the LP relaxation has an optimal solution with all integer values, even if there's another optimal solution, that ties the integer one, with fractional values. Your instance must have $I = J$, that is, all customer nodes are also potential facility sites. Your instance must have at most four nodes.

Include the following in your report:

- A diagram of the nodes and edges.
- The values of h_i, f_j, and c_{ij} for all i, j.
- The optimal solution (x_{LP} and y_{LP}) and optimal objective value (z_{LP}) for the LP relaxation.
- The optimal solution (x^* and y^*) and optimal objective value (z^*) for the IP.

8.15 (LP Relaxation of pMP) Repeat Problem 8.14 but for the pMP instead of the UFLP.

8.16 (Ignoring Some Customers in the UFLP) The UFLP includes a constraint that requires every customer to be assigned to some facility. It is often the case that a small handful of customers in remote regions of the geographical area are difficult to serve and can influence the solution disproportionately. In this problem, you will formulate a version of the UFLP in which a certain percentage of the demands may be ignored when calculating the objective function.

Let α be the minimum fraction of demands to be assigned; that is, a set of customers whose cumulative demand is no more than $100(1 - \alpha)\%$ of the total demand may be ignored. The parameter α is fixed, but the model decides endogenously which customers to ignore. Customers must be either assigned or not—they cannot be assigned fractionally.

a) Using the notation introduced in Section 8.2.2, formulate this problem—we'll call it the "partial assignment UFLP" (PAUFLP)—as a linear integer programming problem. Explain each of your constraints in words.

b) Now consider adding a dummy facility, call it u, to the original UFLP. Facility u has a fixed capacity, so we are really dealing with the capacitated fixed-charge location problem (CFLP), not the UFLP. (See Section 8.3.1 for more on the CFLP.) Assigning customers to this dummy facility in the CFLP represents choosing not to assign them in the PAUFLP. Explain how to set the dummy facility's parameters—its fixed cost, capacity, and transportation cost to each customer—so that solving the CFLP with the dummy facility is equivalent to solving the PAUFLP. Formulate the resulting integer programming problem.

c) Using Lagrangian relaxation, relax the assignment constraints in your model from part (b). Formulate the Lagrangian subproblem, using λ_i as the Lagrange multiplier for the assignment constraint for customer i.

d) Explain how to solve the Lagrangian subproblem you wrote in part (c) for fixed values of λ.

e) Once you have a solution to the Lagrangian subproblem for fixed values of λ, how can you convert it to a feasible solution to the CFLP?

8.17 (UFLP with Multiple Assignments) Suppose that, in the UFLP, customers do not receive 100% of their demand from their nearest open facility. For example, a given customer might receive 80% of its demand from the closest facility, 15% from the second-closest, and 5% from the third-closest. This situation might arise, for example, when locating ambulances, repair centers, or other services for which the primary facility may sometimes be busy.

Let m be the maximum number of facilities that serve each customer, and let b_{ik} be the fraction of demand that customer $i \in I$ receives from the kth-closest open facility, for $k = 1, \ldots, m$. (In the example above, $m = 3$, $b_{i1} = 0.8$, $b_{i2} = 0.15$, and $b_{i3} = 0.05$.) The

b_{ik} are inputs to the model; that is, the assignment fractions are known in advance. Assume that, for a given i, the b_{ik} are nonincreasing in k.

a) Formulate this problem as an integer linear optimization problem. Use the notation introduced in Section 8.2.2, with the following modification: y_{ijk} equals 1 if facility j serves customer i as the kth closest, and 0 otherwise. If you introduce any new notation, define it clearly. Explain the objective function and each constraint in words.

b) If customer i is assigned to j_1 at level k_1 and j_2 at level k_2 for $k_1 < k_2$, then we must have $c_{ij_1} \leq c_{ij_2}$. Explain why the model does *not* need a constraint enforcing this condition.

c) If we require $y_{ijk} \geq 0$ rather than $y_{ijk} \in \{0, 1\}$, as we did in the UFLP, does there always exist an optimal solution in which these variables are binary, as there is in the UFLP?

d) In your model from part (a), you should have a constraint that requires each customer i to be assigned to exactly one facility j at each proximity level k. Relax this constraint via Lagrangian relaxation. Write the Lagrangian subproblem that results. Explain how to solve this problem efficiently for fixed values of the Lagrange multipliers. Your method must be exact (i.e., it must be guaranteed to find the optimal solution) and self-contained (i.e., it may not rely on CPLEX or another solver).

e) *Bonus*: Suppose the b_{ik} are *not* nonincreasing in k. Then the distance-ordering property in part (c) may not hold unless we enforce it using constraints. Write constraints to enforce this condition.

8.18 (Relaxing x Variables in UFLP) Prove or disprove the following claim: If we constrain the y variables to be binary in the UFLP but allow the x variables to be continuous, then there always exists an optimal solution to the resulting problem in which the x variables are binary.

8.19 (Locating Paper Factories) A paper company needs to decide where to locate paper factories in order to supply its five regional branches, which are located in Akron, OH, Albany, NY, Nashua, NH, Scranton, PA, and Utica, NY. The Assistant to the Regional Manager of the Scranton office has selected four potential locations for factories: Bethlehem, PA, Pittsburgh, PA, Rochester, NY, and Springfield, MA. Table 8.3 lists the annual fixed costs and capacities at the four potential plant locations; the annual demand at each of the regional branches; and the cost to produce and ship one case of paper from each plant to each branch. Plant capacities and branch demands are expressed in cases per year.

Where should the company build its plants? Which plant(s) should each branch receive paper from? What is the total cost of your solution? Solve the problem using the modeling environment and solver of your choice.

8.20 (DUALOC #1) Figure 8.23 depicts an instance of the UFLP with three customers (marked as circles) and three potential facility sites (marked as squares). Fixed costs f_j are marked next to each facility. Each customer has a demand of $h_i = 1$, and transportation costs are equal to the Manhattan-metric distance between the facility and customer.

Apply DUALOC's dual-ascent procedure (Algorithm 8.4) to this instance. Report:

- The values of v_i for all $i \in I$ and s_j for all $j \in J$ at the end of the first complete iteration, i.e., after looping through all the customers once.

Table 8.3 Paper-company data for Problem 8.19.

	Production + Shipping Costs				
	Bethlehem	Pittsburgh	Rochester	Springfield	Demand
Akron	\$2.20	\$1.80	\$2.70	\$3.80	1,200,000
Albany	\$1.60	\$3.20	\$1.20	\$0.60	1,150,000
Nasuha	\$3.20	\$4.00	\$2.50	\$0.70	1,350,000
Scranton	\$0.80	\$2.10	\$1.40	\$1.30	1,800,000
Utica	\$1.60	\$2.40	\$0.70	\$1.50	900,000
Fixed Cost	\$4,000,000	\$7,500,000	\$4,500,000	\$5,200,000	
Capacity	3,300,000	4,800,000	4,200,000	3,750,000	

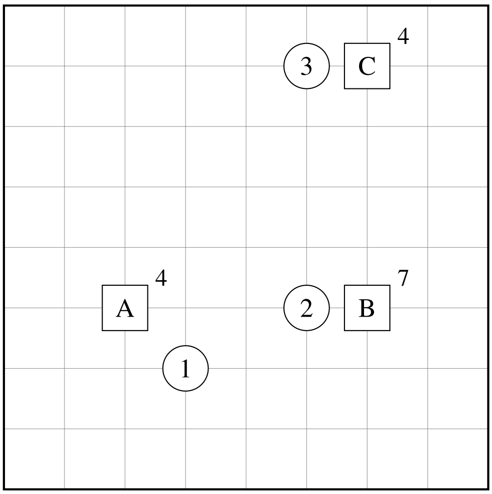


Figure 8.23 UFLP instance for Problem 8.20. Distances use Manhattan metric.

- The final values of v^+, J^+, x^+, y^+, and the dual and primal objective function values.
- Whether the solution to this instance of the UFLP is (a) definitely optimal, (b) definitely sub-optimal, or (c) you can't tell.

8.21 (DUALOC #2) Repeat Problem 8.20 for the instance depicted in Figure 8.24.

8.22 (Warehouses for Quikflix) Quikflix is a mail-order DVD-rental company. You choose which DVDs to rent on Quikflix's web site, and the company mails the DVDs to you. When you've finished watching the movies, you mail them back to Quikflix. Quikflix's business plan depends on fast shipping times (otherwise, customers will get impatient). But overnight delivery services like FedEx are prohibitively expensive. Instead, Quikflix has decided to open enough DCs so that roughly 90% of their customers enjoy 1-day delivery times.

In this problem, you will formulate and solve a model to determine where Quikflix should locate DCs to ensure that a desired percentage of the US population is within a

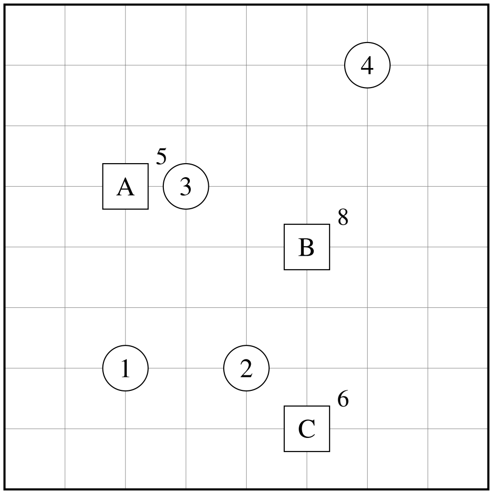


Figure 8.24 UFLP instance for Problem 8.21. Distances use Manhattan metric.

1-day mailing range while minimizing the fixed cost to open the DCs. (You may assume that the per-unit cost of processing and shipping DVDs is the same at every DC.)

a) Formulate the following problem as an integer programming problem: We are given a set of cities, as well as the population of each city and the fixed cost to open a DC in that city. The objective is to decide in which cities to locate DCs in order to minimize the total fixed cost while also ensuring that at least α fraction of the population is within a 1-day mailing range.

Define your notation clearly and indicate which items are parameters (inputs) and which are decision variables. Explain each of your constraints in words.

b) Implement your model using a modeling language of your choice. Solve the problem using the data set provided in `quikflix.xlsx`, which gives the locations and populations of the 250 largest cities in the United States (according to the 2000 US Census), as well as the average annual fixed costs to open a DC in the cities (which are fictitious). The file also contains the distance between each pair of cities in the data set, in miles. Assume that two cities are within a 1-day mailing radius if they are no more than 150 miles apart.

Using these data and a coverage percentage of $\alpha = 0.9$, find the optimal solution to the Quikflix DC location problem. Include a printout of your model file (data not necessary) in your report. Report the total cost of your solution and the total number of DCs open.

8.23 (Solving the Quikflix Problem) In Problem 8.22, you formulated an IP model to solve Quikflix's problem of locating DCs to ensure that a given fraction (α) of the population is within a 1-day mailing range of its nearest DC. In this problem, you will develop a method for solving this IP using Lagrangian relaxation.

The IP formulation for Problem 8.22 contains two sets of decision variables. We'll assume that the x variables represent location decisions, while the z variables indicate whether or not a city is covered (i.e., is within a 1-day mailing radius of an open facility). If you defined z as a continuous variable, make sure you have added a constraint requiring it to be less than or equal to 1. (This constraint is not strictly necessary since it is implied by other constraints, but it strengthens the Lagrangian relaxation formulation.)

The IP formulation also has a set of constraints that allow city i to be covered only if there is an open facility that is less than 150 miles away. If necessary, rewrite your model so that those constraints are written as $\leq$ constraints. Then relax those constraints, and let λ_i be the Lagrange multiplier for the constraint corresponding to node $i \in J$, where J is the set of cities.

a) Write out the Lagrangian subproblem that results from this relaxation.

b) The subproblem should decompose into two separate problems, one containing only the x variables and one containing only the z variables. Write out these two separate problems.

c) Explain how to solve each of the two subproblems, the x-subproblem and the z-subproblem. Your solution method may *not* rely on using the simplex method or any other general-purpose LP or IP algorithm.

d) Suppose that the problem parameters and Lagrange multipliers are given by the following values:

i	f_i	h_i	λ_i
1	100	80	-50
2	100	120	-50
3	100	40	-40
4	100	90	-200

Suppose also that $\alpha = 0.7$ and that node 1 covers nodes 1, 2, 3; node 2 covers nodes 1, 2, and 4; node 3 covers nodes 1 and 3; and node 4 covers nodes 2 and 4.

Determine the optimal values of x and z, as well as the optimal objective value, for this iteration of the Lagrangian subproblem.

8.24 (UFLP with Enemy Customers) Suppose that, in the UFLP model, some pairs of customers are "enemies" and cannot be served by the same facility. Let $a_{ik} = 1$ if customers $i, k \in I$ $(i \neq k)$ are enemies of each other, 0 otherwise. (a_{ik} is a parameter.) Assume that the enemy pairs don't overlap: If i and k are enemies of each other, then i and k aren't enemies of any other customers.

a) Write one or more linear constraints that can be added to the UFLP to enforce the condition that two customers may not be assigned to the same facility if they are enemies of each other. If you introduce any new notation, define it clearly.

b) Suppose we add your constraints from part (a) to the UFLP and then relax constraints (8.4) using Lagrangian relaxation, with Lagrange multipliers λ_i. Write the resulting Lagrangian subproblem.

c) Explain how to solve the Lagrangian subproblem you formulated in part (b) for fixed values of λ. Your solution method may not rely on using the simplex method or any other general-purpose LP or IP algorithm.

d) Choose one option and briefly explain your reasoning: For every instance of the UFLP with enemy constraints, the optimal objective function value will be [$\leq$, $<$, $=$, $>$, $\geq$] the optimal objective function value of the corresponding instance of the classical UFLP.

e) Choose one option and briefly explain your reasoning: For every instance of the UFLP with enemy constraints, the optimal number of open facilities will be [$\leq$, $<$, $=$, $>$, $\geq$] the optimal number of open facilities in the corresponding instance of the classical UFLP.

8.25 (Locating Warehouses for Vandelay Industries) Vandelay Industries manufactures latex products at several plants (whose locations must be chosen from among a set of potential locations) and ships products to customers (whose locations and demands are known). There is a fixed cost to open each plant, and each has a fixed production capacity.

For each unit of demand shipped to a given customer, Vandelay Industries earns a certain amount of revenue. However, the company may choose to satisfy only a part of a given customer's demand, or not to satisfy its demand at all (for example, if it is too expensive to ship to that customer). The only penalty for failing to serve a customer is the lost revenue.

In order to ensure adequate service to customers spread throughout the country, Vandelay Industries also wishes to ensure that no two plants are located less than a certain distance apart.

The company's objective is to maximize the total profit, accounting for the revenue from serving customers and the costs of opening facilities and shipping goods to customers.

Formulate this problem as a linear mixed-integer optimization problem (MIP). In addition to the notation in Sections 8.2.2 and 8.3.1, please use the following notation. If you use any additional notation, define it clearly.

c_{jk} = distance (miles) between plant $j \in J$ and plant $k \in J$
$c_{\min}$ = minimum allowable distance (miles) between two open plants

8.26 (Locating Snack Bars) You have been hired as a consultant for a new theme park to help choose locations for the park's snack bars (restaurants). The park has been divided into sectors, each representing a small area of land. The management team has forecast the number of people that are expected to be in each sector at any point in time.

Let I be the set of sectors and let J be the set of possible locations for the snack bars. The set J is a subset of I because each possible snack bar location is also a sector. Let h_i be the number of people located in sector i, for $i \in I$. (Of course, h_i is just an estimate, because this number will constantly be changing, but we'll treat it as though the number of people in sector i is static and deterministic.) Let t_{ij} be the number of minutes it takes to walk from sector i to sector j.

The management team has decided there will be four snack bars in the theme park. The snack bars are to be located so as to maximize the number of people that are within a 5-minute walk of a snack bar. Let a_{ij} equal 1 if sector j is within a 5-minute walk of sector i; that is,

$$a_{ij} = \begin{cases} 1, & \text{if } t_{ij} \leq 5 \\ 0, & \text{otherwise} \end{cases}$$

Let x_j equal 1 if we locate a snack bar in sector j and 0 otherwise ($j \in J$). Let z_i equal 1 if sector i is within a 5-minute walk of a snack bar ($i \in I$).

a) Formulate this problem as a linear mixed-integer optimization problem (MIP). If you use any new notation, define it clearly. Explain your constraints in words.

b) Suppose that the management team wants instead to maximize the number of customers covered by at least *two* snack bars. We can redefine z_i to equal 1 if sector i is covered by at least two open snack bars. Explain how to modify your model from part (a) to enforce this new requirement. Clearly define any new notation you introduce and explain your new constraint(s) in words.

c) Return to the original formulation—assume again that a customer is "covered" if there is one open snack bar within 5 minutes. Suppose now the management team also wants to ensure that the average distance traveled by a customer to his or her closest snack bar is no more than 6 minutes. (The average is taken across all customers.) That is, we want to maximize the number of customers within 5 minutes of a snack bar, but we also want to ensure that the average time for *all* customers is no more than 6 minutes. Revise the model to include this requirement. Clearly define any new notation you introduce and explain any new constraints in words.

d) Continuing with the model in part (c), suppose that the management wants to require that the average distance traveled by a customer to his or her *second-closest* snack bar is no more than 6 minutes. Explain how to modify your model from part (c) to include this requirement. Clearly define any new notation you introduce and explain any new constraints in words.

8.27 (Locating RFID Readers) The theme park from Problem 8.26 issues bands to all of the visitors to the park. The bands are worn on the wrist, and they contain RFID chips that allow the park to identify visitors, without paper tickets, barcodes, etc. The RFID chips are "read" by RFID readers that are located throughout the park—at the park entrance, near the entrances to rides, and so on. RFID is wireless, and each RFID reader can detect RFID chips that are within a certain radius. In fact, there are two types of RFID readers—short-range and long-range—and the wrist bands contain *both* types of RFID chips. Some locations within the park must be covered by a short-range reader, some by a long-range reader, and some by both.

Two technical constraints restrict the locations of the readers:

1. Short- and long-range readers cannot be placed at the same location.
2. No location can be covered by more than four readers, total (including both types).

Park planners want to locate RFID readers throughout the park to cover all of the necessary sites with the reader types required, at minimum possible cost, while satisfying the technical constraints.

a) Let I be a set of nodes representing locations in the park that must be covered by an RFID reader. (We'll call these "demand nodes.") Let J be a set of nodes representing potential sites for the readers. Let $k = 1, 2$ be the two types of readers (1 = short-range, 2 = long-range). Let r_{ik} be a parameter (an input) that equals 1 if demand node $i \in I$ must be covered by a reader of type k. Let f_{jk} be the fixed cost to locate a type-k reader at location $j \in J$. Let x_{jk} be a decision variable that equals 1 if we locate a reader of type k at location $j \in J$.

Using this notation, formulate the problem as a linear integer optimization model. Explain the objective function and the constraints in words. If you introduce any new notation, define it clearly.

b) Now suppose the theme park's engineers have found a way to locate a short-range and a long-range RFID reader in the same location $j \in J$, but due to the expense involved in doing so, planners wish to have at most two locations that have both types of readers. Write one or more linear constraints to enforce this restriction.

8.28 (Locating Compost Sites) The city of Greentown is planning to open several composting facilities, which will convert organic matter (kitchen waste, leaves, yard waste, shredded paper, etc.) into fertilizer instead of sending it to landfills. While the population of Greentown agrees that this is a good idea, nobody wants a new compost site too close to their homes, due to the noise, smell, and truck traffic to and from the site. The city's mayor has hired you to develop a model to choose locations for the new compost facilities.

The population of the city has been aggregated into a set I of neighborhoods, each with population h_i. City planners have identified a set J of potential sites for the compost facilities. The distance between neighborhood i and site j is given by c_{ij} miles. The city wishes to locate p compost sites in order to *maximize the minimum distance between a neighborhood and its nearest open compost facility*.

Define the following decision variables:

$$x_j = \begin{cases} 1, & \text{if we locate a compost facility at site } j, \\ 0, & \text{otherwise} \end{cases}$$

$$y_{ij} = \begin{cases} 1, & \text{if site } j \text{ is the nearest open compost facility to neighborhood } i, \\ 0, & \text{otherwise} \end{cases}$$

a) Formulate this problem as a linear integer optimization model. Explain the objective function and constraints in words as well as formulating them in mathematical notation. If you introduce any new notation (sets, parameters, decision variables), define it clearly.

b) Now suppose that, instead of maximizing the minimum distance between a neighborhood and its nearest open facility, the mayor wants to *maximize the shortest distance between any two open compost facilities*. Note that this objective function focuses only on the distances among compost facilities and ignores distances between facilities and neighborhoods.

Formulate this modified problem as a linear integer optimization model. Explain the objective function and new constraints in words. If you introduce any new notation, define it clearly.

8.29 (Convex Hulls are Nonoverlapping) Consider a facility location instance with nodes in $\mathbb{R}^2$ and Euclidean distances. Suppose we open a set $J' \subseteq J$ of facilities and assign each customer in I to the nearest open facility. Recall that the *neighborhood* of an open facility j is $N_j \equiv \{i \in I | y_{ij=1}\}$. Prove that the convex hulls of the neighborhoods of the open facilities do not overlap.

8.30 (LR Iteration for UFLP) The file `LR-UFLP.xlsx` contains data for a 50-node instance of the UFLP, as well as the Lagrange multipliers for a single iteration of the Lagrangian relaxation algorithm described in Section 8.2.3. For each facility $j \in J$, column `B` lists the fixed cost f_j. For each customer $i \in I$, row `2` lists the demand h_i and row `3` lists the Lagrange multiplier λ_i. Finally, the cells in the range `C6:AZ55` contain the matrix of transportation costs c_{ij}.

a) For each $j \in J$, calculate the benefit β_j, the optimal value of x_j, and the optimal objective value of (UFLP-LR$_\lambda$). The worksheet labeled "`solution`" contains spaces to list β_j (column `B`), x_j (column `C`), and the objective value (cell `C5`).

Hint: To double-check your calculations, we'll tell you that if $i = 6$ and $j = 3$, then $h_i c_{ij} - \lambda_i = 12422.34$.

b) Using the method described in Section 8.2.3.4, generate a feasible solution to the UFLP. In row `2` of the "`solution`" worksheet, list the index of the facility that each customer is assigned to in your solution. In cell `C6`, list the objective value of your solution.

8.31 (Maxisum Location Problem) Consider the following problem: We must locate exactly p facilities, for fixed p. The objective is to *maximize* the sum of the demand-weighted distances between each customer and its nearest facility. Formulate this problem as an IP. Define any new notation clearly. Explain the objective function and each of the constraints in words.

8.32 (Supplier–Facility Capacities) Consider the following extension of the UFLP: We are given a set K of suppliers whose locations are fixed. Each supplier $k \in K$ can ship at most b_{jk} units to facility $j \in J$. This is like a capacity constraint, but it is (supplier, facility)-specific rather than the facility-specific capacities discussed in Section 8.3.1. Such constraints might arise from, say, the capacity of the truck transporting goods from k to j. Let d_{jk} be the cost to transport one unit of demand from supplier $k \in K$ to facility $j \in J$, and let z_{jk} be a decision variable representing the number of units transported from k to j. Note that z_{jk} is a flow-type variable ($z_{jk} \geq 0$), whereas y_{ij} is a fractional variable ($0 \leq y_{ij} \leq 1$). Multiple-sourcing is allowed; that is, facility j may receive shipments from more than one supplier k. In addition to the notation just defined, use the notation in Section 8.2.2. If you need to define any additional notation, define it clearly.

a) Formulate this extension of the UFLP as a linear mixed-integer optimization problem. Explain the objective function and each constraint clearly in words.

b) In Section 8.2.3, we solved the UFLP by relaxing the "assignment" constraints that require each customer to be assigned to exactly 1 facility. Write the objective function of the Lagrangian subproblem that results from relaxing the analogous constraint in your model from part (a).

c) Consider the special case in which $h_i = h$ for all $i \in I$, i.e., all of the customers have the same demand, and b_{jk} is an integer multiple of h for all j, k. Explain how to solve the Lagrangian subproblem from part (b) for this special case. Your method must be exact (i.e., it must be guaranteed to find the optimal solution) and self-contained (i.e., it may not rely on a general-purpose optimization solver).

d) Describe a method that, given a feasible solution to the Lagrangian subproblem, produces a feasible solution for the original problem.

8.33 (Salt Stockpiles) You are the director of your local Department of Transportation. You have decided to build silos to stockpile the salt the department uses on roadways during winter weather. A stockpile is considered to cover a town if they are within r miles of each other. Your job is to determine where to locate up to p stockpiles to maximize the total population of the towns that are *double-covered*, i.e., covered by at least *two* stockpiles. Local planners have provided you with the population of each town that you would like to be covered.

a) Formulate this problem as an integer programming problem. Define any new notation clearly.

b) Now suppose that the two stockpiles that double-cover a given town must be *at least* s miles from each other. (Two stockpiles may be *located* less than s miles from each other, but a given town doesn't count as double-covered unless there are two stockpiles that cover it and that are at least s miles apart.) Formulate the new model, and define any new notation clearly.

8.34 (Pre-Positioning Disaster Relief Shelters) A disaster relief agency plans to establish shelters in preparation for a hurricane that has been forecast for the coming days. The agency wishes to choose shelters from a set J of potential locations in order to cover every population center in the set I. A shelter covers a population center if it is within r miles of it. As in the set covering and maximal covering models, we define the parameter a_{ij} to equal 1 if a shelter at site $j \in J$ covers population center $i \in I$.

If we locate a shelter at site j, we incur a fixed cost of f_j, as well as an "assignment cost" of w_j for each population center assigned to the shelter at j (regardless of the size of these population centers). For example, if shelter j serves 12 population centers, then we pay an assignment cost of $12w_j$.

Define the following decision variables:

$$x_j = \begin{cases} 1, & \text{if we locate a shelter at site } j \\ 0, & \text{otherwise} \end{cases}$$

$$y_{ij} = \begin{cases} 1, & \text{if a shelter at site } j \text{ serves population center } i \\ 0, & \text{otherwise} \end{cases}$$

a) Formulate this problem as a linear integer optimization problem. If you introduce any new notation, define it clearly. Briefly explain your objective function and constraints.

b) In part (a), the assignment cost is a linear function of the number of population centers assigned to each shelter: It equals $w_j n$, where n is the number of population centers assigned to j. Suppose instead that the assignment cost is a *nonlinear* function $g_j(n)$, where n is the number of population centers assigned to j. Define the following decision variables:

$$z_{jn} = \begin{cases} 1, & \text{if exactly } n \text{ population centers are assigned to a shelter at } j \\ 0, & \text{otherwise} \end{cases}$$

Formulate this problem as a linear integer programming problem. Define any new notation clearly, and explain the objective function and any new constraints.

8.35 (Stochastic Pre-Positioning) A humanitarian relief agency wishes to pre-position stockpiles of emergency supplies (food, water, blankets, medicine, etc.) for use in the aftermath of disasters. Its objective is to locate the smallest possible number of stockpiles while ensuring a low probability that, for each population center, a disaster strikes and the population center cannot be served by any stockpile. Whether a given stockpile can serve a given population center depends on their physical distance as well as on the disaster that strikes.

Disasters are represented by scenarios. A scenario can be thought of as a disaster type, magnitude, and location (e.g., magnitude 7.5 earthquake in city A, influenza pandemic in city B, etc.). However, mathematically each scenario simply specifies whether a given population center can be served by a given stockpile during a given disaster.

Let I be the set of population centers, and let J be the set of potential stockpile locations. Let S be the set of scenarios (including the scenario in which no disaster occurs), and let q_s be the probability that scenario s occurs. Stockpile j is said to "cover" population center i in scenario s if *either* stockpile j can serve population center i in scenario s *or* population center i does not need disaster relief in scenario s. Let a_{ijs} be a parameter that equals 1 if stockpile j covers population center i in scenario s, and 0 otherwise. Assume each stockpile is sufficiently large to serve the needs of the entire population it covers.

Formulate a linear integer programming problem that chooses where to locate stockpiles in order to minimize the total number of stockpiles located while ensuring that, for each $i \in I$, the probability that i is not covered by any open stockpile is less than or equal to α, for given $0 \le \alpha \le 1$. Clearly define any new notation you introduce. Explain the objective function and all constraints in words.

8.36 (Error Bias) Suppose the transportation costs are estimated badly in the UFLP. It is natural to expect that the true cost of the solution found under the erroneous data has an equal probability of being larger or smaller than the cost calculated when solving the problem. Test this hypothesis by solving the instance given in `random-errors.xlsx` 100 times, each time perturbing the transportation costs by multiplying them by $U[0.75, 1.25]$ random variates. For each instance generated this way, record the objective function value, as well as the objective function of the same solution when the correct costs are used. If the hypothesis is correct, the objective function should be less than the true cost for roughly half of the instances and greater for the other half. Do your results confirm the hypothesis? In a few sentences, explain your results, and why they occurred. Also comment on the implications your results have for the importance of having accurate data when choosing facility locations.

8.37 (1-Center on a Tree) Consider the 1-center problem on a tree network in which all of the demands are 1. Prove that the Algorithm 8.10 finds the optimal solution to both the absolute and the vertex 1-center problem. (Recall from Section 8.4.3 that the *absolute p-center problem* allows facilities to be located on either the edges or the nodes of the network, whereas the *vertex p-center problem* restricts facilities to the nodes.)

Algorithm 8.10 1-Center on a tree

$v_1 \leftarrow$ any point on the tree
$v_2 \leftarrow$ node that is farthest from v_1
$v_3 \leftarrow$ node that is farthest from v_2
absolute 1-center is at the midpoint of the (unique) path from v_2 to v_3; vertex 1-center is at the vertex of the tree that is closest to the absolute 1-center

8.38 (2-Center on a Tree) Prove that Algorithm 8.11 finds the optimal solution to the absolute 2-center problem.

8.39 (N-Echelon Location Problem) By extending the approach used in Section 8.7.1, formulate a facility location model with N echelons, for general $N \ge 3$. Echelon N ships

Algorithm 8.11 2-Center on a tree

1: using Algorithm 8.10, find the absolute 1-center of the tree
2: delete from the tree the link containing the absolute 1-center. (If the absolute 1-center is on a vertex, delete one of the links incident to the center on the path from v_1 to v_2.) This divides the tree into two disconnected subtrees
3: use Algorithm 8.10 to find the absolute 1-center of each of the subtrees; these constitute a solution to the absolute 2-center problem

products to echelon $N-1$, which ships products to echelon $N-2$, and so on; echelon 1 serves the end customer. The locations of the facilities in echelons $2, \ldots, N$ are to be decided by the model, and there are fixed costs for each. Define any new notation clearly. Explain the objective function and each of the constraints in words. *Note*: No decision variables should have more than 3 indices.

8.40 (UFLP Duality Gap) Prove Lemma 8.4.

8.41 (Another Relaxation for the pMP) Suppose that we use Lagrangian relaxation to relax constraint 8.72 in the pMP. Write the resulting Lagrangian subproblem. This problem is structurally identical to another problem discussed in this chapter; what is it? Briefly summarize the advantages and disadvantages of this relaxation compared to the relaxation discussed in Section 8.3.2.2: Which subproblem is harder to solve? Which approach will give a tighter bound? For which approach will the subgradient optimization procedure converge more quickly?

8.42 (Tightening the CFLP Relaxation) Suppose we add the following constraint to the CFLP:

$$\sum_{j \in J} v_j x_j \geq \sum_{i \in I} h_i. \tag{8.152}$$

Explain in words what this constraint says. Explain why this constraint is redundant for the CFLP (adding it does not change the optimal solution for the CFLP) and why adding it tightens the Lagrangian relaxation discussed in Section 8.3.1. Finally, explain how to solve the Lagrangian subproblem when constraint (8.152) is included in the model.

8.43 (Variable-Splitting Method for CFLP) In this problem, you will develop a *variable-splitting* method for the CFLP. Variable splitting (also known as *Lagrangian decomposition*) is a method that involves duplicating one or more sets of variables, adding a constraint that requires those variables to be equal to their duplicates, and then relaxing that constraint using Lagrangian relaxation. (See Guignard and Kim (1987).)

a) Introduce new decision variables w_{ij} for $i \in I$, $j \in J$. Rewrite the objective function as

$$\text{minimize} \quad \sum_{j \in J} f_j x_j + \beta \sum_{i \in I} \sum_{j \in J} h_i c_{ij} y_{ij} + (1-\beta) \sum_{i \in I} \sum_{j \in J} h_i c_{ij} w_{ij},$$

where $0 \leq \beta \leq 1$ is a constant. Rewrite constraints (8.55) using w instead of y. Add the following new constraints, which require w and y to be equal, and require w to be nonnegative:

$$w_{ij} = y_{ij} \qquad \forall i \in I, \forall j \in J \tag{8.153}$$

$$w_{ij} \geq 0 \qquad \forall i \in I, \forall j \in J \tag{8.154}$$

Write the resulting problem. This problem is equivalent to (CFLP).

b) Relax constraints (8.153) using Lagrangian relaxation. Write the resulting subproblem.

c) Explain how to solve the subproblem from part (b).

d) Based on your intuition, will this relaxation provide a tighter, weaker, or equivalent bound to the relaxation discussed in Section 8.3.1?

8.44 (Accuracy of Spherical Law of Cosines Formula) Calculate the distances between every pair of nodes in the 88-node data set (`88node.xlsx`) using both the great circle distance formula (8.1) and the spherical law of cosines formula (8.2). Compare the results. Are there any cases for which the two formulas produce distances that differ by more than a mile or so? If so, what characterizes those cases?

8.45 (Swap vs. Neighborhood Search for p-Median) Implement the swap and neighborhood search heuristics for the pMP (Algorithms 8.7 and 8.8). Conduct a numerical experiment to compare the effectiveness (as measured by objective function value) and efficiency (as measured by CPU time) of these two heuristics. Your experiment should use randomly generated p-median instances with at least 100 nodes.

8.46 (Hakimi Property for SCLP) Does the Hakimi property hold for the set covering problem? Explain your answer.

8.47 (Proof of Lemma 8.8) Prove Lemma 8.8.

8.48 (Solving the pCP using the MCLP) Algorithm 8.9 relies on the relationship between the pCP and the SCLP stated in Lemma 8.8. A similar relationship exists between the pCP and the MCLP.

a) State a lemma similar to Lemma 8.8 that describes this relationship.

b) Write pseudocode similar to Algorithm 8.9 for an exact algorithm that solves the pCP by iteratively solving MCLPs.

8.49 (The MCLP with Mandatory Closeness Constraints) The *MCLP with mandatory closeness constraints* is identical to the MCLP except that it also requires *every* customer to be covered within a distance of s, with $s \geq r$. That is, we wish to maximize the number of demands that are covered within r, but every customer must be covered within s. Write an integer programming formulation for this problem. If you introduce any new notation, define it clearly. Explain the objective function and each constraint in words.

8.50 (MCLP is a Special Case of pMP) Show that the MCLP is a special case of the pMP by showing how to set the parameters of the pMP so that solving it is equivalent to solving the MCLP.

8.51 (A Dynamic Location Problem) Consider a dynamic facility location problem in which the demands over a finite time horizon are known but change in each time period: h_{it} is the demand at node $i \in I$ in period t, where $t = 1, \ldots, T$. We can open and close as many facilities as we like in each time period. Facility $j \in J$ incurs a fixed cost of f^+_{jt} if it is opened in period t (but was closed in period $t-1$), a fixed cost of f^-_{jt} if it is closed in period t (but was open in period $t-1$), and a fixed cost of f_{jt} if it remains open in period t (that is, if it was also open in period $t-1$). Assume that no facilities are open at the start of the horizon, that is, in period 0. The transportation cost from facility j to customer i in period t is given by c_{ijt}. Formulate an integer programming model to optimize the locations of facilities over the time horizon to minimize the total fixed and transportation costs. If you introduce any new notation, define it clearly. Explain your objective function and constraints in words.

8.52 (MCLP Modifications) Modify the MCLP to accommodate each of the changes described below (one at a time). For each modification, change *either* the objective function *or* exactly one constraint to reflect the modification. Indicate the number of the equation (objective function or constraint) you are changing.

a) We wish to maximize the total number of nodes covered, not the total population covered.

b) Each facility j has a fixed construction cost of f_j. Rather than restricting the number of facilities to equal p, restrict the total amount spent to construct facilities to a budget of b.

c) A demand node only counts as covered if there are *two* facilities within the coverage radius.

8.53 (Subproblem Assignments) Prove that, if customer i is assigned to at least one facility in the optimal solution to (UFLP-LR$_\lambda$), then one of the facilities it is assigned to is the nearest open facility. (This implies that in step 4 of Algorithm 8.2, it suffices to check only those j such that $y_{ij} = 1$ in the optimal solution to (UFLP-LR$_\lambda$).)

8.54 (Location of Power Generators) Consider the problem of locating generators within an electricity network.

a) First consider a single generator. Suppose the generator's load (i.e., the total demand for electricity from the generator) is given by $D \sim N(\mu, \sigma^2)$, where D is measured in kilowatt-hours (kWh). The cost to generate enough electricity to meet a load of d kWh is given by $\frac{1}{2}\gamma d^2$, where $\gamma > 0$ is a constant. Prove that the expected generation cost is given by $\frac{1}{2}\gamma(\mu^2 + \sigma^2)$.

b) Now consider an electricity network consisting of multiple generators, whose locations we need to choose. Let I be the set of loads (demand nodes), with load i having a daily demand distributed $N(\mu_i, \sigma_i^2)$. Let J be the set of potential generators. The daily fixed cost if generator j is open is f_j, and the generation cost coefficient for j is γ_j. Formulate the problem of choosing generator locations and assigning loads to generators in order to minimize the expected daily cost of the system. Assume that, once location and assignment decisions are made, the power network for a given generator and its loads is disconnected from the remaining generators and loads (so that the physics of power flows can be ignored). Also assume that the cost to transmit power is negligible.

8.55 (Stochastic Location for Toy Stores) Return to Problem 8.1, and suppose now that the demands are stochastic. The file `toy-stores-stochastic.xlsx` gives the demands for five scenarios, as well as the probability that each scenario occurs.

a) Implement the stochastic fixed-charge location problem in a modeling language of your choice. Find the optimal solution for the instance given in the data set. Report the optimal set of facilities and the corresponding cost.

b) Now implement and solve the minimax fixed-charge location problem. Report the optimal set of facilities and the corresponding cost.

8.56 (Minimax Cost $\neq$ Minimax Regret) Construct a small example of the minimax fixed-charge location problem (MFLP) in which minimizing the maximum cost results in an optimal solution that is different from the solution that minimizes the maximum regret. (You may choose either relative or absolute regret.) Your instance may have at most five nodes.

8.57 (Side Constraints for Arc Design) Formulate each side constraint listed below for the arc design model in Section 8.7.2.2. Your constraints must be linear. If you introduce any new notation, define it clearly.

a) We have a set $P \subseteq E \times E$ of ordered pairs of arcs such that, for $(e_1, e_2) \in P$, if arc e_1 is opened, then arc e_2 must be opened.

b) We have a set of $E' \subseteq E$ of arcs such that at most r arcs in E' may be opened.

c) We have a set of $E' \subseteq E$ of arcs such that at least r arcs in E' must be opened.

d) We have an upper bound B on the transportation cost that may be spent shipping on a subset $E' \subseteq E$ of the arcs.

8.58 (Modified Hungary Network) Consider the Hungary instance of the arc design problem shown in Figure 8.20. The file `hungary2.xlsx` contains a modification of the instance described in Example 8.11. It lists the latitude and longitude of each node, the available units for each node and product, and the fixed cost and capacity for each arc. The variable cost is 1 for every arc and product. Formulate the arc design model in a modeling language of your choice, and solve this instance. Report the optimal arcs to open, the optimal flows, and the optimal total cost.

8.59 (Campaign Offices) A candidate for a national political position wishes to establish campaign offices and decide how much money to spend on campaign activities at those offices. The candidate's staff has identified a set J of potential locations for campaign offices (facilities) and a set I of neighborhoods (demand nodes) that they wish to "cover" using these offices. Let a_{ij} be a parameter that equals 1 if office location $j \in J$ covers neighborhood $i \in I$, and 0 otherwise. Neighborhood $i \in I$ has h_i registered voters living in it. Opening an office at location $j \in J$ incurs a fixed cost of f_j.

In addition to choosing *where* to locate offices, the candidate's staff needs to determine how much money to spend on campaign activities (get-out-the-vote, marketing, etc.) at each office. They can only perform campaign activities at offices that they have chosen to open. Staffers have estimated that each \$1 spent on these activities will earn the candidate exactly one extra vote.

For example: Suppose the candidate opens an office at location $j \in J$, and location j covers 1000 registered voters. If the campaign spends \$1000 on campaign activities (*not* including the fixed cost f_j), the candidate will earn all of their votes; if it spends \$500, the candidate will earn half of their votes; and if it spends \$0, the candidate will earn none of their votes. Note that there is no advantage to spending more than \$1000 on campaign activities in this example. There is also no advantage to opening an office at j if we spend \$0 since the candidate will not earn any votes.

If a neighborhood is covered by more than one open campaign office, its votes can only be earned once. Therefore, only one office should direct its campaign activities at that neighborhood. Your model should choose which of the open offices should "serve" each customer.

The candidate's objective is to maximize the number of votes earned. The campaign has a total budget of $\$B$ to spend on *both* fixed costs *and* campaign activities.

Define the following decision variables:

x_j = 1, if we open a campaign office at location $j \in J$, 0 otherwise
w_j = the number of dollars we spend on campaign activities at office $j \in J$

Formulate this problem as an integer linear optimization problem. If you introduce any new notation, define it clearly. Explain your objective function and each constraint in words.

8.60 (Exchange Rate Hedging) An automobile manufacturer wishes to decide where to locate factories around the world in order to account for random fluctuations in currency exchange rates. The company will change the production levels at the various factories to take advantage of changes in the exchange rates. Exchange rates are expressed as α \$/¤, where \$ stands for US dollars (USD) and ¤ stands for the local currency in the other country. For example, if the exchange rate between the United States and Thailand is $\alpha = 0.028$ \$/฿, then 1 Thai baht is worth US\$0.028.

The manufacturer is considering a set J of potential locations for the factories, which will ship automobiles directly to the customers in a set I. Customer $i \in I$ has a demand of h_i units per year. We have the following costs:

- Building a factory at site $j \in J$ incurs a fixed annual cost of $\$f_j$, which is deterministic and expressed in USD.
- The cost to produce one automobile at factory $j \in J$ is ¤b_j, which is deterministic and expressed in the local currency of the country in which factory j is located.
- The cost to ship one automobile from factory $j \in J$ to customer $i \in I$ is $\$c_{ij}$, which is deterministic and expressed in USD.

The factories have effectively unlimited capacity.

Once the factories are built, the random exchange rates are realized, and the company then decides how much to produce at each factory, as well as how much to ship from each factory to each customer. The exchange rates are described by a set S of scenarios, such that α_{js} is the exchange rate (in \$/¤) in scenario s for the country in which facility $j \in J$ is located. Let q_s be the probability that scenario s occurs.

Let x_j equal 1 if we open a factory at site $j \in J$, 0 otherwise. Let y_{ijs} equal the number of automobiles to be shipped from a factory at site $j \in J$ to customer $i \in I$ in scenario $s \in S$. These are our decision variables. You may treat y_{ijs} as a continuous variable.

a) Formulate a stochastic optimization problem that minimizes the total expected annual cost of locating facilities and producing and transporting automobiles. If you introduce any new notation, define it clearly. Explain your objective function and each constraint in words.

b) Suppose we allow y_{ijs} to be continuous and nonnegative. If the demands h_i are expressed as integers, will there necessarily exist an optimal solution in which the y_{ijs} are integers? Why or why not?

c) Suppose that, instead of minimizing the total expected cost, the company wishes to minimize the maximum absolute regret that can occur, across all exchange rate scenarios. Formulate this new problem. If you introduce any new notation, define it clearly.

CHAPTER 9

SUPPLY UNCERTAINTY

9.1 INTRODUCTION TO SUPPLY UNCERTAINTY

Supply chains are subject to many types of uncertainty, and many approaches have been proposed for modeling uncertainty in the supply chain. So far in this book, we have primarily considered uncertainty in demand. In this chapter, we study models that consider uncertainty in supply; in other words, what happens when a firm's suppliers, or the firm's own facilities, are unreliable.

Supply uncertainty may take a number of forms. These include:

- *Disruptions.* A disruption interrupts the supply of goods at some stage in the supply chain. Disruptions tend to be binary events—either there's a disruption or there isn't. During a disruption, there's generally no supply available. Disruptions may be due to bad weather, natural disasters, strikes, suppliers going out of business, etc.

- *Yield Uncertainty.* Sometimes the quantity that a supplier can provide falls short of the amount ordered; the amount actually supplied is random. This is called yield uncertainty. It can be the result of product defects, or of batch processes in which only a certain percentage of a given batch (the yield) is usable.

- *Capacity Uncertainty.* Uncertainty in the quantity that a supplier can provide. Whereas yield uncertainty is typically dependent on the order quantity (e.g., we order S units, but only a portion of them are usable), capacity uncertainty usually as-

Fundamentals of Supply Chain Theory, Second Edition. Lawrence V. Snyder and Zuo-Jun Max Shen.

Companion website: www.wiley.com/go/Snyder/SupplyChainTheory

sumes the capacity is independent of the order quantity, and the supplier will deliver the minimum of the two.

- *Lead Time Uncertainty.* Uncertainty in the supply lead time can result from stockouts at the supplier, manufacturing or transit delays, and so on. In this case, the lead time L that figures into many of the models in this book must be treated as a random variable rather than a constant. See, for example, Section 5.3.3.

In this chapter, we will discuss the first two types of supply uncertainty. We will discuss models for setting inventory levels in the presence of disruptions in Section 9.2 and in the presence of yield uncertainty in Section 9.3. In both sections, we will cover models that are analogous to the classical economic order quantity (EOQ) and infinite-horizon newsvendor models (the models from Sections 3.2 and 4.3.4). We discuss a newsvendor-type model with a more general supply process in Section 9.4. Next, we discuss the risk-diversification effect, a supply-uncertainty version of the risk-pooling effect, in Section 9.5. Finally, in Section 9.6, we discuss a facility location model with supply uncertainty in the form of disruptions.

In most of the models in this chapter, we will assume that demand is deterministic. We do this for tractability, but also, more importantly, to highlight the effect of supply uncertainty, in the absence of demand uncertainty.

In some ways, there is no conceptual difference between supply uncertainty and demand uncertainty. After all, having too little supply is the same as having too much demand. A firm might use similar strategies for dealing with the two types of uncertainty, as well—for example, holding safety stock, utilizing multiple suppliers, or improving its forecasts of the uncertain events. But, as we will see, the ways in which we model these two types of uncertainty, and the insights we get from these models, can be quite different. (For more on this issue, see Snyder and Shen (2006).)

For reviews of the literature on disruptions, see Snyder et al. (2016) and Vakharia and Yenipazarli (2008), and for yield uncertainty, see Yano and Lee (1995) and Grosfeld Nir and Gerchak (2004). Ciarallo et al. (1994) discuss capacity uncertainty. For an overview of models with lead-time uncertainty, see Zipkin (2000).

9.2 INVENTORY MODELS WITH DISRUPTIONS

Disruptions are usually modeled using a two-state Markov process in which one state represents the supplier operating normally and the other represents a disruption. These states may be known as up/down, wet/dry, on/off, normal/disrupted, and so on. (We'll use the terms up/down.) Not surprisingly, continuous-review models (such as the one in Section 9.2.1) use continuous-time Markov chains (CTMCs), while periodic-review models (Section 9.2.2) use discrete-time Markov chains (DTMCs). The time between disruptions and the length of disruptions are therefore exponentially or geometrically distributed (in the case of CTMCs and DTMCs, respectively). The models presented here assume the inventory manager knows the state of the supplier at all times.

Some papers also consider more general disruption processes than the ones we consider here—for example, nonstationary disruption probabilities (Snyder and Tomlin 2007) or partial disruptions (Güllü et al. 1999). These disruption processes can also usually be modeled using Markov processes.

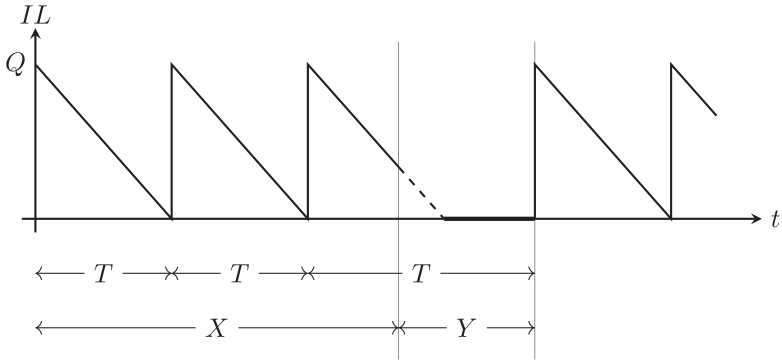


Figure 9.1 EOQ inventory curve with disruptions.

9.2.1 The EOQ Model with Disruptions

9.2.1.1 Problem Statement Consider the classical EOQ model with fixed order cost K and holding cost h per unit per year. The demand rate is d units per year (a change from our notation in Section 3.2). Suppose that the supplier is not perfectly reliable—that it functions normally for a certain amount of time (an up interval) and then shuts down for a certain amount of time (a down interval). The transitions between these intervals are governed by a CTMC. During down intervals, no orders can be placed, and if the retailer runs out of inventory during a down interval, all demands observed until the beginning of the next up interval are lost, with a stockout cost of p per lost sale. During up intervals, the lead time is 0. Both types of intervals last for a random amount of time. Every order placed by the retailer is for the same fixed quantity Q. Our goal is to *choose Q to minimize the expected annual cost*.

This problem, which is known as the *EOQ with disruptions* (EOQD), was first introduced by Parlar and Berkin (1991), but their analysis contained two errors that rendered their model incorrect. A correct model was presented by Berk and Arreola-Risa (1994), whose treatment we follow here.

Let X and Y be the duration of a given up and down interval, respectively. X and Y are exponentially distributed random variables, X with rate λ and Y with rate μ. (Recall that if $X \sim \exp(\lambda)$, then $f(x) = \lambda e^{-\lambda x}$, $F(x) = 1 - e^{-\lambda x}$, and $\mathbb{E}[X] = 1/\lambda$.) The parameters λ and μ are called the *disruption rate* and *recovery rate*, respectively. These are the transition rates for the CTMC.

The EOQ inventory curve now looks something like Figure 9.1. Note that the inventory position never becomes negative because excess demands are lost, not backordered. The time between successful orders is called a *cycle*. The length of a cycle, T, is a random variable. If the supplier is in an up interval when the inventory level reaches 0, then $T = Q/d$, otherwise, $T > Q/d$.

Note: In the EOQ, we ignored the per-unit purchase cost c because the annual per-unit cost is independent of Q (since d units are ordered every year, regardless of Q). It is not strictly correct to ignore c in the EOQD because, in the face of lost sales, the number of units ordered each year may not equal d, and in fact it depends on Q. Nevertheless, we will ignore c for tractability reasons.

9.2.1.2 Expected Cost Let ψ be the probability that the supplier is in a down interval when the inventory level hits 0. One can show that

$$\psi = \frac{\lambda}{\lambda + \mu}\left(1 - e^{-\frac{(\lambda+\mu)Q}{d}}\right). \tag{9.1}$$

Let $f(t)$ be the pdf of T, the time between successful orders. Then

$$f(t) = \begin{cases} 0, & \text{if } t < Q/d \\ 1 - \psi, & \text{if } t = Q/d \\ \psi\mu e^{-\mu(t-Q/d)}, & \text{if } t > Q/d. \end{cases}$$

Note that $f(t)$ has a point mass at Q/d and is continuous afterwards.

Each cycle lasts at least Q/d years. After that, with probability $1-\psi$, it lasts an additional 0 years, and with probability ψ, it lasts, on average, an additional $1/\mu$ years (because of the memoryless property of the exponential distribution). Therefore, the expected length of a cycle is given by

$$\mathbb{E}[T] = \frac{Q}{d} + \frac{\psi}{\mu}. \tag{9.2}$$

We're interested in finding an expression for the expected cost per year. It's difficult to write an expression for this cost directly. On the other hand, we can calculate the expected cost of one cycle, as well as the expected length of a cycle, and the time between orders is iid. This implies that we can use the renewal-reward theorem (Theorem 4.7), treating each successful order as a renewal point. In particular, the renewal-reward theorem tells us that the expected cost per year, $g(Q)$, is given by

$$g(Q) = \frac{\mathbb{E}[\text{cost per cycle}]}{\mathbb{E}[\text{cycle length}]}. \tag{9.3}$$

The denominator is given by (9.2); it remains to find an expression for the numerator.

In each cycle, we place exactly one order, incurring a fixed cost of K. The inventory in a given cycle is positive for exactly Q/d years (regardless of whether there's a disruption), so the holding cost is based on the area of one triangle in Figure 9.1, namely $Q^2/2d$. Finally, we incur a penalty cost if the supplier is in a down interval when the inventory level hits 0. This happens with probability ψ, and if it does happen, the expected remaining duration of the down interval is $1/\mu$. Therefore, the expected stockout cost per cycle is $pd\psi/\mu$. Then the total expected cost per cycle is

$$K + \frac{hQ^2}{2d} + \frac{pd\psi}{\mu}. \tag{9.4}$$

We can use (9.2)–(9.4) to derive the expected cost per year; the result is stated in the next proposition.

Proposition 9.1 *In the EOQD, the expected cost per year is given by*

$$g(Q) = \frac{K + hQ^2/2d + pd\psi/\mu}{Q/d + \psi/\mu}. \tag{9.5}$$

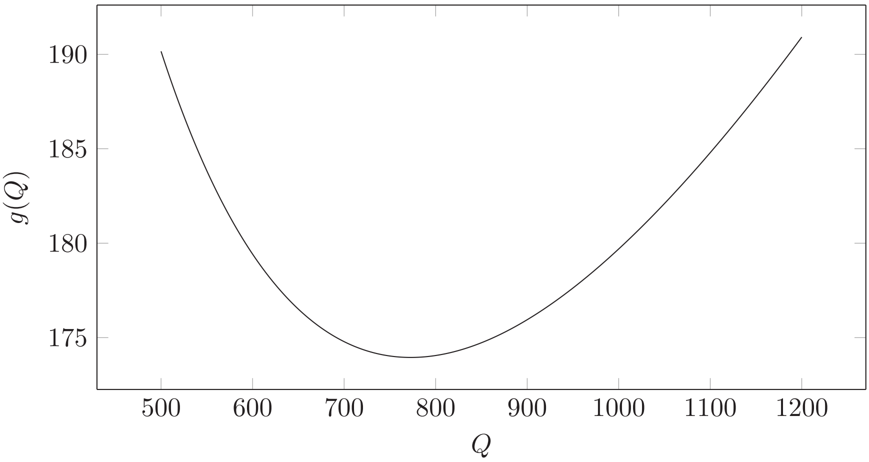


Figure 9.2 Exact EOQD cost $g(Q)$ for Example 9.1.

9.2.1.3 Solution Method Remember that ψ is a function of Q, and in fact it's a pretty messy function of Q. Therefore, (9.5) can't be solved in closed form—that is, we can't take a derivative, set it equal to 0, and solve for Q. Instead, it must be solved numerically using line search techniques such as bisection search. These techniques typically assume that $g(Q)$ is convex. Unfortunately, it is not known whether $g(Q)$ is convex with respect to Q, but it is known that $g(Q)$ is *quasiconvex* in Q. A quasiconvex function has only one local minimum, which is a sufficient condition for most line search techniques to work.

□ **EXAMPLE 9.1**

Recall Joe's Corner Store from Example 3.1. Suppose that Joe's supplier is subject to disruptions, with up and down intervals that have exponentially distributed durations with rates 1.5 and 14, respectively. (That is, disruptions begin, on average, $1/1.5 = 0.6667$ years after the last disruption ended, and they last, on average, $1/14 = 0.0714$ years, or 0.8571 months.) Recall that $d = 1300$, $K = 8$, and $h = 0.225$ and suppose that, as in Example 3.8, $p = 5$. What is the optimal order quantity and the corresponding expected cost?

Figure 9.2 plots $g(Q)$. We optimized $g(Q)$ numerically and found that $Q^* = 772.81$ and $g(Q^*) = 173.95$. Note that the optimal order is nearly twice the size of the order placed when there are no disruptions, in Example 3.1. The cost is considerably higher, too.

□

There's nothing wrong with solving the EOQD numerically, insofar as the algorithm for doing so is quite efficient. On the other hand, it's desirable to have a closed-form solution for it for two main reasons. One is that we may want to embed the EOQD into some larger model rather than implementing it as-is. (See, e.g., Qi et al. (2010).) Doing so may require a closed-form expression for the optimal solution or the optimal cost. The other reason is that we can often get insights from closed-form solutions that we can't get from solutions we have to obtain numerically.

Although we can't get an exact solution for the EOQD in closed form, we can get an approximate one. In particular, Snyder (2014) approximates ψ by ignoring the exponential term:

$$\hat{\psi} = \frac{\lambda}{\lambda + \mu}. \tag{9.6}$$

$\hat{\psi}$ is the probability that the supplier is in a down interval at an arbitrary point in time. But ψ refers to a specific point in time, i.e., the point when the inventory level hits 0, and the term $(1 - e^{-(\lambda+\mu)Q/d})$ in the definition of ψ accounts for the knowledge that, when this happens, we were in an up interval Q/d years ago.

By replacing ψ with $\hat{\psi}$, then, we are essentially assuming that the system approaches steady state quickly enough that when the inventory level hits 0, we can ignore this bit of knowledge, i.e., ignore the transient nature of the system at this moment. The approximation is most effective, then, when cycles tend to be long; e.g., when Q/d is large. If Q/d is large, then $(\lambda + \mu)Q/d$ is large, $e^{-(\lambda+\mu)Q/d}$ is small, and $\hat{\psi} \approx \psi$. The approximation tends to be quite tight for reasonable values of the parameters.

The advantage of using $\hat{\psi}$ in place of ψ is that the resulting expected cost function no longer has any exponential terms, and we can set its derivative to 0 and solve for Q in closed form. (See Problem 9.7(b).) This also allows us to perform some of the same analysis on the EOQD that we do on the EOQ—for example, we can perform sensitivity analysis, develop worst-case bounds for power-of-two policies, and so on. It also allows an examination of the cost of using the classical EOQ solution when disruptions are possible; as it happens, the cost of this error can be quite large.

☐ **EXAMPLE 9.2**

Figure 9.3 plots both the exact cost function, $g(Q)$, and the approximate cost function, $\hat{g}(Q)$, for the problem in Example 9.1. The two curves are virtually indistinguishable. Using $\hat{\psi}$ in place of ψ and optimizing numerically (or using the closed-form expression in Problem 9.7(b)), we get $\hat{Q}^* = 773.14$—very close to the exact Q^*, 772.81, in Example 9.1. Similarly, we get $\hat{g}(\hat{Q}^*) = 173.96$, whereas $g(Q^*) = 173.95$ in Example 9.1. Note that $e^{-(\lambda+\mu)Q^*/d} = 9.96 \times 10^{-5}$, confirming the claim above that this term tends to be small.

☐

9.2.2 The Newsvendor Problem with Disruptions

In this section, we consider the infinite-horizon newsvendor problem of Section 4.3.4, except that in place of demand uncertainty, we have supply uncertainty, in the form of disruptions. We know from Section 4.3.4 that in the case of demand uncertainty, a base-stock policy is optimal, with the optimal base-stock level given by

$$S^* = \mu + \sigma\Phi^{-1}\left(\frac{p}{p+h}\right) \tag{9.7}$$

(if demand is normally distributed and $\gamma = 1$). We will see that the optimal solution for the problem with supply uncertainty has a remarkably similar form.

The model we discuss below can be viewed as a special case of models introduced by Güllü et al. (1997) and by Tomlin (2006). Elements of our analysis are adapted from

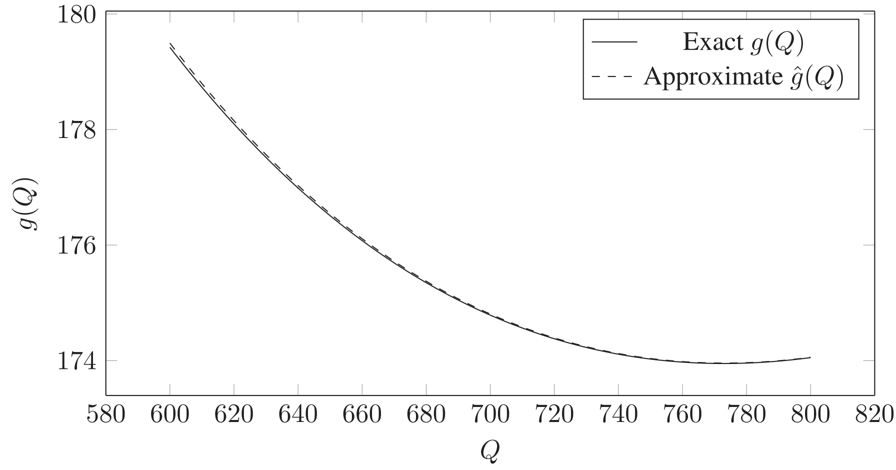


Figure 9.3 Approximate and exact EOQD costs $\hat{g}(Q)$ and $g(Q)$ for Example 9.2.

Tomlin (2006) and from the unabridged version of that paper (Tomlin 2005). Some of the analysis can also be found in Schmitt et al. (2010).

9.2.2.1 Problem Statement

As in Section 9.2.1 on the EOQD, we assume that demand is deterministic; it's equal to d units per period. (d need not be an integer.) On-hand inventory and backorders incur costs of h and p per unit per period, respectively. There is no lead time. The sequence of events is identical to that described in Section 4.3, except that in step 2, no order is placed if the supplier is disrupted.

The probability that the supplier is disrupted in the next period depends on its state in the current period. In other words, the disruption process follows a two-state DTMC. Let

$$\alpha = \mathbb{P}(\text{down next period}|\text{up this period})$$
$$\beta = \mathbb{P}(\text{up next period}|\text{down this period}).$$

We refer to α as the *disruption probability* and β as the *recovery probability*. These are the transition probabilities for the DTMC. The up and down periods both constitute geometric processes, and these processes are the discrete-time analogues to the continuous-time up/down processes in Section 9.2.1.

Given the transition probabilities α and β, we can solve the steady-state equations to derive the steady-state probabilities of being in an up or down state as follows:

$$\pi_u = \frac{\beta}{\alpha+\beta} \tag{9.8}$$
$$\pi_d = \frac{\alpha}{\alpha+\beta} \tag{9.9}$$

It turns out to be convenient to work with a more granular Markov chain that indicates not only whether the supplier is in an up or down period, but also how long the current down interval has lasted. In particular, state n in this Markov chain represents being in a down interval that has lasted for n consecutive periods. If $n = 0$, we are in an up period.

Let π_n be the steady-state probability that the supplier is in a disruption that has lasted n periods. Furthermore, define

$$F(n) = \sum_{i=0}^{n} \pi_i. \tag{9.10}$$

$F(n)$ is the cdf of this process and represents the steady-state probability that the supplier is in a disruption that has lasted n periods or fewer (including the probability that it is not disrupted at all). These probabilities are given explicitly in the following lemma, but often, we will ignore the explicit form of the probabilities and just use π_n and $F(n)$ directly.

Lemma 9.2 *If the disruption probability is α and the recovery probability is β, then*

$$\begin{aligned}
\pi_0 &= \frac{\beta}{\alpha+\beta} \\
\pi_n &= \frac{\alpha\beta}{\alpha+\beta}(1-\beta)^{n-1}, \quad n \geq 1 \\
F(n) &= 1 - \frac{\alpha}{\alpha+\beta}(1-\beta)^n, \quad n \geq 0.
\end{aligned}$$

Proof. Omitted; see Problem 9.10. ■

9.2.2.2 Form of the Optimal Policy Our objective is to make inventory decisions to minimize the expected holding and stockout cost per period. What type of inventory policy should we use? It turns out that a base-stock policy is optimal for this problem:

Theorem 9.3 *A base-stock policy is optimal in each period of the infinite-horizon newsvendor problem with deterministic demand and stochastic supply disruptions.*

We omit the proof of Theorem 9.3; it follows from a much more general theorem proved by Song and Zipkin (1996). Note that a base-stock policy works somewhat differently in this problem than in previous problems, since we might not be able to order up to the base-stock level in every period—in particular, we can't order *anything* during down periods. So a base-stock policy means that we order up to the base-stock level during up periods and order nothing during down periods. The extra inventory during up periods is meant to protect us against down periods.

9.2.2.3 Expected Cost Suppose the supplier is in state $n = 0$; that is, an up period. If we order up to a base-stock level of S at the beginning of the period, we incur a cost at the end of the period of

$$h(S-d)^+ + p(d-S)^+. \tag{9.11}$$

In state $n = 1$, we incur a cost of

$$h(S-2d)^+ + p(2d-S)^+, \tag{9.12}$$

and in general, we incur a cost of

$$h\,[S-(n+1)d]^+ + p\,[(n+1)d-S]^+ \tag{9.13}$$

in state n, for $n = 0, 1, \ldots$.

Therefore, the expected holding and stockout costs per period can be expressed as a function of S as follows:

$$g(S) = \sum_{n=0}^{\infty} \pi_n \left[h\left[S - (n+1)d\right]^+ + p\left[(n+1)d - S\right]^+ \right]. \tag{9.14}$$

9.2.2.4 Optimal Solution

Lemma 9.4 *The optimal base-stock level S^* is an integer multiple of d.*

Proof (sketch). The proof follows from the fact that g is a piecewise-linear function of S, with breakpoints at multiples of d. ■

Normally, we would find the optimal S by taking a derivative of $g(S)$, but since S is discrete (by Lemma 9.4), we need to use a *finite difference* instead, as we did for the newsvendor problem with a discrete demand distribution in Section 4.3.2.8. In particular, S^* is the smallest S that is an integer multiple of d such that $\Delta g(S) \geq 0$, where

$$\Delta g(S) = g(S+d) - g(S). \tag{9.15}$$

(In Section 4.3.2.8, we defined $\Delta g(S)$ as $g(S+1) - g(S)$, but here, since S can only take on values that are multiples of d, it's sufficient to define $\Delta g(S)$ as in (9.15).)

$$\begin{aligned}
\Delta g(S) &= g(S+d) - g(S) \\
&= \sum_{n=0}^{\infty} \pi_n \Big[h\left[S - nd\right]^+ + p\left[nd - S\right]^+ \\
&\qquad\qquad - h\left[S-(n+1)d\right]^+ - p\left[(n+1)d - S\right]^+ \Big]
\end{aligned}$$

Now,

$$[S - nd]^+ - [S - (n+1)d]^+ = \begin{cases} d, & \text{if } n < \frac{S}{d} \\ 0, & \text{otherwise} \end{cases}$$

and

$$[nd - S]^+ - [(n+1)d - S]^+ = \begin{cases} -d, & \text{if } n \geq \frac{S}{d} \\ 0, & \text{otherwise.} \end{cases}$$

Therefore,

$$\begin{aligned}
\Delta g(S) &= d\left[h \sum_{n=0}^{\frac{S}{d}-1} \pi_n - p \sum_{n=\frac{S}{d}}^{\infty} \pi_n \right] \\
&= d\left[hF\left(\frac{S}{d} - 1\right) - p\left(1 - F\left(\frac{S}{d} - 1\right)\right)\right] \\
&= d\left[(h+p)F\left(\frac{S}{d} - 1\right) - p\right],
\end{aligned}$$

where F is as defined in (9.10). Then S^* is the smallest multiple of d such that

$$(h+p)F\left(\frac{S}{d} - 1\right) - p \geq 0 \tag{9.16}$$

$$\Longleftrightarrow S \geq d + dF^{-1}\left(\frac{p}{p+h}\right), \tag{9.17}$$

where $F^{-1}(\gamma)$ is interpreted as the smallest n such that $F(n) \geq \gamma$. Interpreted this way, $F^{-1}(\gamma)$ is an integer for all γ, the right-hand side of (9.17) is automatically a multiple of d, and we can drop the "smallest multiple of d" language and replace the inequality in (9.17) with an equality.

We have now proved the following:

Theorem 9.5 *In the infinite-horizon newsvendor problem with deterministic demand and stochastic supply disruptions, the optimal base-stock level is given by*

$$S^* = d + dF^{-1}\left(\frac{p}{p+h}\right), \tag{9.18}$$

where F is as defined in (9.10) *and F^{-1} is interpreted as described above.*

Notice that the optimal base-stock level under supply uncertainty has a very similar structure to that under demand uncertainty, as given in (9.7). First, it uses the familiar newsvendor critical ratio $p/(p+h)$, but here the inverse cdf F^{-1} refers not to the demand distribution but to the supply distribution.

Second, the right-hand side of (9.18) has a natural cycle stock–safety stock interpretation, just like in the demand uncertainty case. Here, d is the cycle stock—the inventory to meet this period's demand—and $dF^{-1}(\gamma)$, where $\gamma = p/(p+h)$, is the safety stock—the inventory to protect against uncertainty (in this case, supply uncertainty).[1]

Just like in the demand uncertainty case, the optimal solution specifies what fractile of the distribution we should protect against. Here, we should have enough inventory to protect against any disruption whose length is no more than $F^{-1}(\gamma)$ periods. The probability of a given period being in a disruption that has lasted longer than this is $1 - \gamma$, so, as in the demand uncertainty case, the type-1 service level is given by γ. As usual, the base-stock level increases with p and decreases with h.

☐ **EXAMPLE 9.3**

Gauss & Poisson (G&P; see Example 7.1) relies on a certain unreliable supplier for a key raw material used to make toothpaste. G&P produces 2000 cases of toothpaste per day. Each case of toothpaste carried in inventory incurs a holding cost of \$0.25, and each case of toothpaste that cannot be manufactured because of a lack of raw materials incurs a stockout cost of \$3.00 per day. The supplier has a disruption probability of $\alpha = 0.04$ and a recovery probability of 0.25. (Thus, disruptions occur, on average, every 25 days and last, on average, 4 days.) What is the base-stock level should G&P use to manage the inventory of the raw material, and what is the optimal expected cost per day?

First, note that this example treats the manufacturing process as the "demand," and the inventory is of raw materials used in the manufacturing process, rather than of finished goods. We have $d = 2000$, $h = 0.25$, $p = 3$. Using Lemma 9.2, we get the pmf π_n and cdf $F(n)$ shown in Table 9.1 for $0 \leq n \leq 10$. (Ignore the last column for now.)

[1] In earlier chapters, we used $\alpha = p/(p+h)$; here, we use γ since α has a new meaning in this section.

Table 9.1 pmf, cdf, and costs of supplier disruptions in Example 9.3.

n	π_n	$F(n)$	$\pi_n \left[h\,[S-(n+1)d]^+ + p\,[(n+1)d - S]^+ \right]$
0	0.8621	0.8621	1293.10
1	0.0345	0.8966	34.48
2	0.0259	0.9224	12.93
3	0.0194	0.9418	0.00
4	0.0145	0.9564	87.28
5	0.0109	0.9673	130.93
6	0.0082	0.9755	147.29
7	0.0061	0.9816	147.29
8	0.0046	0.9862	138.09
9	0.0035	0.9896	124.28
10	0.0026	0.9922	108.74

We have $\gamma = p/(p+h) = 0.9231$, so $F^{-1}(\gamma) = 3$, interpreting $F^{-1}(\gamma)$ as the smallest integer n such that $F(n) \geq \gamma$. Thus, by Theorem 9.5, $S^* = 2000 + 2000 \cdot 3 = 8000$. In other words, we should hold enough raw materials to cover us for a disruption of up to 3 days; after that, we will begin to stock out.

The last column of Table 9.1 gives the summand of (9.14) for $0 \leq n \leq 10$. For example, if $n = 0$ (we are in a nondisrupted period), we will end the period with 6,000 units on hand, for a cost of $0.25 \cdot 6{,}000 = 1{,}500$, and this occurs with probability 0.8621, so the 0th term in the sum in (9.14) is $0.8621 \cdot 1500 = 1293.10$. Similarly, if $n = 5$ (we are in the fifth period of a disruption), then we will end the period with 4000 units of backorders, for a cost of $0.0109 \cdot 3 \cdot 4000 = 130.93$. The total cost (approximated by calculating the summands through $n = 100$) is $g(S^*) = 2737.07$. □

9.3 INVENTORY MODELS WITH YIELD UNCERTAINTY

In some cases, the number of items received from the supplier may not equal the number ordered. This may happen because of stockouts or machine failures at the supplier, or because the production process is subject to defects. The quantity actually received is called the *yield*. If the yield is deterministic—e.g., we always receive 80% of our order size—then the problem is easy: we just multiply our order size by $1/0.8 = 1.25$. More commonly, however, there is a significant amount of uncertainty in the yield. The optimal solution under *yield uncertainty* generally involves increasing the order quantity, as under imperfect but deterministic yield, but it should account for the variability in yield, not just the mean—just as in the case of demand uncertainty.

In the sources of yield uncertainty mentioned above, we'd expect that the actual yield should always be less than or equal to the order quantity—we shouldn't receive more than we order. But yield uncertainty can also occur in batch production processes—e.g., for chemicals or pharmaceuticals—or in agriculture. In this case, it's not a matter of items being "defective," but rather of not knowing in advance precisely how much usable product

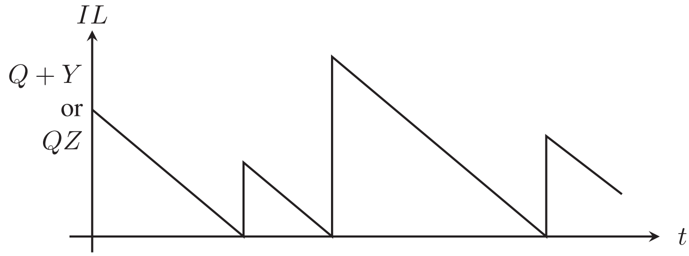


Figure 9.4 EOQ inventory curve with yield uncertainty.

will result from the process. The amount received may therefore be more than the amount expected, and we can't necessarily place an upper bound on the yield.

In this section, we consider how to set inventory levels under yield uncertainty. As in Section 9.2, we consider both a continuous-review setting, based on the EOQ model, and a periodic-review setting, based on the newsvendor problem. As before, we will assume that demand is deterministic.

There are many ways to model yield uncertainty. We will consider two that are intuitive and tractable.

The first is an *additive yield uncertainty* model in which we assume that if an order of size Q is placed, then the yield (the amount received) equals $Q + Y$. Y is a continuous random variable with pdf f_Y and cdf F_Y. Y need not be normal, or even symmetric. Y might be bounded from above by 0 if the yield can never exceed the order quantity; in this case, it might have an point mass at 0 (otherwise, the yield would equal 100% with 0 probability). Typically, the yield distribution is truncated at $-Q$ (since we can't receive a negative amount), but we'll use $-\infty$ as its lower bound, primarily because it's inconvenient to have the yield distribution depend on the order size.

The second approach is a *multiplicative yield uncertainty* approach in which the yield is given as QZ, where Z is a continuous, nonnegative random variable with pdf f_Z and cdf F_Z. Again, Z need not be symmetric. If the yield cannot exceed Q, then $Z \leq 1$.

In both cases, we assume that the yield distribution (f_Y or f_Z) does not depend on Q. This assumption may or may not be realistic; it is made primarily for mathematical convenience.

9.3.1 The EOQ Model with Yield Uncertainty

9.3.1.1 Problem Statement The setup for this problem is just like the EOQ model, except that if an order is placed for Q units, the actual number of units received may differ from Q. Unlike the EOQD in Section 9.2.1, the supplier never experiences down intervals, so it's always possible to place an order, even if the quantity delivered falls short of the quantity ordered. That means that, unlike the EOQD, we never have stockouts in the EOQ with yield uncertainty. (See Figure 9.4.)

As in the EOQD, we'll derive the expected cost per year as a function of Q using the renewal-reward theorem. Here, we can define a renewal simply as an order. We need to derive expressions for the expected cost per cycle and the expected cycle length.

9.3.1.2 Additive Yield Let's first consider the additive yield approach, in which the yield is given by $Q + Y$. In each cycle, we place exactly one order, so the fixed order cost is given by K. The expected holding cost is given by h times the area of one triangle in Figure 9.4, but these triangles have varying heights and widths. In particular, if the yield is $Q + Y$, then the holding cost is $h(Q + Y)^2/2d$. Therefore, the expected cost per cycle is given by

$$\begin{aligned}
K + \int_{-\infty}^{\infty} \frac{h}{2d}(Q+y)^2 f_Y(y)dy =& K + \frac{h}{2d}\left[Q^2 \int_{-\infty}^{\infty} f_Y(y)dy \right. \\
&\left. +2Q\int_{-\infty}^{\infty} y f_Y(y)dy + \int_{-\infty}^{\infty} y^2 f_Y(y)dy\right] \quad (9.19) \\
=& K + \frac{h}{2d}\left[Q^2 + 2Q\mathbb{E}[Y] + \mathbb{E}[Y^2]\right] \\
=& K + \frac{h}{2d}\left[Q^2 + 2Q\mathbb{E}[Y] + \mathrm{Var}[Y] + \mathbb{E}[Y]^2\right] \\
=& K + \frac{h}{2d}\left[(Q + \mathbb{E}[Y])^2 + \mathrm{Var}[Y]\right]. \quad (9.20)
\end{aligned}$$

The expected cycle length is given by

$$\frac{Q + \mathbb{E}[Y]}{d}. \quad (9.21)$$

Using the renewal-reward theorem (Theorem 4.7), the total expected cost per year is then

$$g(Q) = \frac{2Kd + h\left[(Q + \mathbb{E}[Y])^2 + \mathrm{Var}[Y]\right]}{2(Q + \mathbb{E}[Y])} = \frac{2Kd + h\mathrm{Var}[Y]}{2(Q + \mathbb{E}[Y])} + \frac{h(Q + \mathbb{E}[Y])}{2}. \quad (9.22)$$

$g(Q)$ is clearly convex with respect to Q, so we can find a minimum by setting its derivative to 0:

$$\begin{aligned}
\frac{dg}{dQ} &= -\frac{2Kd + h\mathrm{Var}[Y]}{2(Q + \mathbb{E}[Y])^2} + \frac{h}{2} = 0 \\
&\implies h(Q + \mathbb{E}[Y])^2 = 2Kd + h\mathrm{Var}[Y] \\
&\implies Q + \mathbb{E}[Y] = \sqrt{\frac{2Kd + h\mathrm{Var}[Y]}{h}} \\
&\implies Q^* = \sqrt{\frac{2Kd}{h} + \mathrm{Var}[Y]} - \mathbb{E}[Y] \quad (9.23)
\end{aligned}$$

Note that if $\mathrm{Var}[Y] = 0$ (i.e., the yield differs from the order quantity but is no longer uncertain), then the solution is equivalent to the classical EOQ solution shifted by $\mathbb{E}[Y]$—i.e., order $\sqrt{2Kd/h}$, but if we will always receive 20 units fewer than we order ($\mathbb{E}[Y] = -20$), then add 20 units to our order. If, in addition $\mathbb{E}[Y] = 0$, then we have the EOQ solution precisely.

Notice also that the optimal solution does not depend on the *distribution* of Y, only its first two moments. The optimal order quantity increases with $\mathrm{Var}[Y]$ but decreases with $\mathbb{E}[Y]$, since we need to over-order less if the additive term is greater.

☐ **EXAMPLE 9.4**

Many vaccines are manufactured by injecting the target virus into chicken eggs, where the virus replicates and is eventually harvested and purified. The yield of this process is stochastic due to uncertainty in the growth rates of the virus inside the eggs, as well as contamination by bacteria and other sources.

Consider a childhood vaccine whose demand is steady at a rate of 75,000 doses per month. If the manufacturer initiates a batch with the intention of producing Q doses, the actual number of doses produced is $Q + Y$, where $Y \sim N(-15{,}000, 9{,}000^2)$. Note that there is a small (5%) chance that *more* than Q doses will be produced, but in most cases $Y < 0$. Each production batch costs the manufacturer \$18,500, and each dose of finished vaccine in inventory incurs a holding cost of \$0.06 per month.

What are the optimal batch size and expected cost per year? On average, how often will the manufacturer produce new batches of the vaccine?

We have $d = 75{,}000$, $K = 18{,}500$, $h = 0.06$, $\mathbb{E}[Y] = -15{,}000$, and $\mathrm{Var}[Y] = 9{,}000^2$. Therefore, by (9.23),

$$Q^* = \sqrt{\frac{2 \cdot 18{,}500 \cdot 75{,}000}{0.06} + 9{,}000^2} - (-15{,}000) = 230{,}246.37.$$

Plugging Q^* into (9.22) (or using Problem 9.13), we get

$$g(Q^*) = 12{,}914.78.$$

From (9.21), the expected cycle length is

$$\frac{230{,}246.37 + (-15{,}000)}{75{,}000} = 2.87,$$

so the manufacturer produces batches approximately every 2.87 months.

☐

9.3.1.3 Multiplicative Yield Now consider the multiplicative yield approach, in which the yield is given by QZ. In analogy to (9.19), the expected cost per cycle is

$$K + \frac{hQ^2}{2d}\int_0^\infty z^2 f_Z(z)dz = K + \frac{hQ^2}{2d}\left(\mathrm{Var}[Z] + \mathbb{E}[Z]^2\right).$$

Similarly, the expected cycle length is $Q\mathbb{E}[Z]/d$, so the expected cost per year is

$$g(Q) = \frac{Kd}{Q\mathbb{E}[Z]} + \frac{hQ(\mathrm{Var}[Z] + \mathbb{E}[Z]^2)}{2\mathbb{E}[Z]}. \tag{9.24}$$

Again, we take a derivative with respect to Q:

$$\frac{dg}{dQ} = -\frac{Kd}{Q^2\mathbb{E}[Z]} + \frac{h(\mathrm{Var}[Z] + \mathbb{E}[Z]^2)}{2\mathbb{E}[Z]} = 0$$

$$\Longrightarrow \quad Q^* = \sqrt{\frac{2Kd}{h(\mathrm{Var}[Z] + \mathbb{E}[Z]^2)}} \tag{9.25}$$

Similar to the additive yield case, the optimal solution reduces to the EOQ solution, scaled by $1/\mathbb{E}[Z]$, if $\mathrm{Var}[Z] = 0$. If, in addition, $\mathbb{E}[Z] = 1$, then we have the EOQ solution exactly.

Here, too, the optimal solution depends only on the first two moments of Z, not its distribution. As before, Q^* decreases with $\mathbb{E}[Z]$, but here it also decreases with $\mathrm{Var}[Z]$. This is somewhat strange behavior. The explanation lies in what Yano and Lee (1995) call the "portfolio effect," which basically means that if the yield is very variable, it's preferable to use smaller batches to increase our chances of getting a "good" batch the next time.

☐ **EXAMPLE 9.5**

Return to Example 9.4 and suppose now that the yield is multiplicative, with the number of doses produced equal to QZ, where $Z \sim \mathrm{Beta}(5,1)$. What are the optimal batch size and expected cost per year? On average, how often will the manufacturer produce new batches of the vaccine?

A $\mathrm{Beta}(\alpha,\beta)$ random variable has a mean of $\alpha/(\alpha+\beta)$, or 0.8333, and a variance of $\alpha\beta/((\alpha+\beta)^2(\alpha+\beta+1))$, or 0.0198. Therefore, by (9.25),

$$Q^* = \sqrt{\frac{2 \cdot 18{,}500 \cdot 75{,}000}{0.06(0.0198 + 0.8333^2)}} = 254{,}460.21.$$

Plugging Q^* into (9.24) (or using Problem 9.14), we get

$$g(Q^*) = 13{,}086.53.$$

The expected cycle length is

$$\frac{Q^*\mathbb{E}[Z]}{d} = \frac{254{,}460.21 \cdot 0.8333}{75{,}000} = 2.83,$$

so the manufacturer produces batches approximately every 2.83 months. ☐

9.3.2 The Newsvendor Problem with Yield Uncertainty

9.3.2.1 Problem Statement Next, we consider the same infinite-horizon newsvendor-type problem as in Section 9.2.2, except that the supplier suffers from yield uncertainty rather than disruptions. As before, we assume that the demand is deterministic and equal to d per period.

9.3.2.2 Additive Yield If we choose a base-stock level of S, then we have $S+Y$ on hand after the shipment arrives but before demand is realized, and the inventory level at the end of the period is $S+Y-d$. This inventory level is positive if $Y > d-S$ and negative otherwise. Therefore, the expected cost per period is given by

$$g(S) = h\int_{d-S}^{\infty}((S+y)-d)f_Y(y)dy + p\int_{-\infty}^{d-S}(d-(S+y))f_Y(y)dy. \qquad (9.26)$$

We can convert this to a newsvendor function by letting $R \equiv d - S$. ($-R$ represents the safety stock: the amount ordered in excess of the demand to protect against yield uncertainty.) Equation (9.26) can then be written as

$$g(R) = p \int_{-\infty}^{R} (R - y) f_Y(y) dy + h \int_{R}^{\infty} (y - R) f_Y(y) dy. \tag{9.27}$$

This equation is identical in form to (4.3) (but note the reversal of the cost coefficients). Therefore, using (4.17) we know that

$$R^* = F_Y^{-1}\left(\frac{h}{h+p}\right),$$

so

$$S^* = d - F_Y^{-1}\left(\frac{h}{h+p}\right). \tag{9.28}$$

Note that the critical ratio has h in the numerator, not p. If $\mathbb{E}[Y] \le 0$ (as is typical), and if $h < p$ (as is also typical), then $F_Y^{-1}(h/(h+p)) < 0$, so (9.28) instructs us to order more than d to compensate for the yield uncertainty. (Even if $\mathbb{E}[Y] > 0$, $F_Y^{-1}(h/(h+p))$ may still be negative, depending on $h/(h+p)$.)

If Y is normally distributed, then

$$\begin{aligned} S^* &= d - \left[\mathbb{E}[Y] + \Phi^{-1}\left(\frac{h}{h+p}\right)\sqrt{\mathrm{Var}[Y]}\right]. \\ &= d - \mathbb{E}[Y] + z_\alpha\sqrt{\mathrm{Var}[Y]} \end{aligned} \tag{9.29}$$

since $z_\alpha = -z_{1-\alpha}$ (C.11). Again, S^* decreases with $\mathbb{E}[Y]$. If $h < p$, then $z_\alpha > 0$, so, like the EOQ model with additive yield in Section 9.3.1.2, S^* increases with $\mathrm{Var}[Y]$.

In both (9.28) and (9.29), the term subtracted from d is a newsvendor quantity ((4.17) or (4.24)) in which the probability distribution function models the supply uncertainty rather than the demand uncertainty (and the critical ratio is reversed).

□ EXAMPLE 9.6

Consider now the influenza (flu) vaccine, which is primarily used during a single season but has a long manufacturing lead time leading up to that season and therefore can be modeled as a newsvendor problem. (In contrast, childhood vaccines like those discussed in Example 9.4 are used at more or less constant rates throughout the year.)

Suppose that a certain manufacturer of influenza vaccine expects a demand of 1.5 million doses this year. A manufacturing batch intended to produce Q units will actually produce $Q + Y$ units, where $Y \sim U[-500{,}000, 500{,}000]$. Unmet demands are lost at a cost of \$75 per dose, and unused doses must be discarded, incurring a cost of \$15 per dose in wasted material. (This is a single-period newsvendor problem, but the results in this section, which assume an infinite horizon, still apply.)

What are the optimal batch size and expected cost per year?

For simplicity, let's use units of 1 million doses. Then $d = 1.5$, $p = 75{,}000{,}000$, $h = 15{,}000{,}000$, $h/(h+p) = 0.1667$. Furthermore,

$$\begin{aligned} f_Y(y) &= 1 \\ F_Y(y) &= \frac{y + 0.5}{1} \end{aligned}$$

for $y \in [-0.5, 0.5]$ and

$$F_Y^{-1}(\gamma) = \gamma - 0.5$$

for $\gamma \in [0, 1]$. Therefore, by (9.28),

$$S^* = 1.5 - (0.1667 - 0.5) = 1.8333$$

and $R^* = 1.5 - 1.8333 = -0.3333$. Therefore, the manufacturer should produce 1.83 million doses of the vaccine, with a safety stock of 0.33 million. From (9.26), we have

$$\begin{aligned} g(R^* = -0.3333) =& 75{,}000{,}000 \int_{-0.5}^{-0.3333} (-0.3333 - y)dy \\ &+ 15{,}000{,}000 \int_{-0.3333}^{0.5} (y + 0.3333)dy \\ =& 75{,}000{,}000 \cdot 0.0139 + 15{,}000{,}000 \cdot 0.3472 \\ =& 6{,}250{,}500. \end{aligned}$$

□

9.3.2.3 Multiplicative Yield

We will only consider a single-period version of the newsvendor problem with multiplicative yield. The multiperiod problem is much more difficult than the version with additive yield. This is because it is more difficult to calculate the inventory level after the shipment arrives but before the demand occurs. In the additive yield model, this simply equalled $S + Y$, but under multiplicative yield, it equals $x + (S - x)Z$, where x is the ending inventory level in the previous period. This dependence on the system state in the previous period complicates the multiperiod analysis significantly.

In the single-period model, assume that we begin with an inventory level of 0, and we order S units. The inventory level after the shipment arrives is therefore SZ, where Z is the random variable representing the yield. The expected cost in the period is given by

$$g(S) = h \int_{d/S}^{\infty} (Sz - d) f_Z(z)dz + p \int_0^{d/S} (d - Sz) f_Z(z)dz. \tag{9.30}$$

Taking the derivative using Leibniz's rule (C.49), we get

$$\begin{aligned} g'(S) =& h \int_{\frac{d}{S}}^{\infty} z f_Z(z)dz - p \int_0^{d/S} z f_Z(z)dz \\ =& h \left[\int_0^{\infty} z f_Z(z)dz - \int_0^{d/S} z f_Z(z)dz \right] - p \int_0^{d/S} z f_Z(z)dz \\ =& h\mathbb{E}[Z] - (h + p) \int_0^{d/S} z f_Z(z)dz. \end{aligned} \tag{9.31}$$

Moreover,

$$g''(S) = -(h + p) \frac{d}{S} f_Z\left(\frac{d}{S}\right) > 0,$$

so $g(S)$ is convex and the first-order condition is sufficient. Setting $g'(S) = 0$, we find that S^* satisfies

$$\int_0^{d/S^*} z f_Z(z) dz = \frac{h\mathbb{E}[Z]}{h+p} \tag{9.32}$$

or, using (C.70),

$$\frac{d}{S^*} F_Z\left(\frac{d}{S^*}\right) - \bar{n}_Z\left(\frac{d}{S^*}\right) = \frac{h\mathbb{E}[Z]}{h+p}. \tag{9.33}$$

Unfortunately, there is no closed-form expression for S^*, but we can solve (9.32) or (9.33) numerically to get S^*.

☐ **EXAMPLE 9.7**

Suppose that the flu vaccine described in Example 9.6 instead exhibits multiplicative yield uncertainty: The yield Z is normally distributed with a mean of 0.8 and a standard deviation of 0.04. (We can treat the probability that $Z > 1$ as negligible.) What are the optimal batch size and expected cost per year?

Recall that $d = 1.5$ and $h/(h+p) = 0.1667$. By (9.33), since $Z \sim N(0.8, 0.04^2)$, the optimal base-stock level satisfies

$$\frac{1.5}{S}\Phi\left(\frac{\frac{d}{S} - 0.8}{0.04}\right) - 0.04\bar{\mathscr{L}}\left(\frac{\frac{d}{S} - 0.8}{0.4}\right) = 0.1667 \cdot 0.8 = 0.1334,$$

using (C.32). Solving numerically we find that $S^* = 1.9669$ does the trick:

$$\begin{aligned}&\frac{1.5}{1.9669}\Phi\left(\frac{\frac{d}{1.9669} - 0.8}{0.04}\right) - 0.04\bar{\mathscr{L}}\left(\frac{\frac{d}{1.9669} - 0.8}{0.4}\right)\\ =&\frac{1.5}{1.9669} \cdot 0.1750 - 0.04 \cdot 0.0038 = 0.1334.\end{aligned}$$

Therefore, the manufacturer should produce 1.97 million doses of the vaccine. From (9.30), we have

$$\begin{aligned}g(1.9669) =& 15{,}000{,}000 \int_{1.5/1.9669}^{\infty} (1.9669z - 1.5) f_Z(z) dz\\ &+ 75{,}000{,}000 \int_0^{1.5/1.9669} (1.5 - 1.9669z) f_Z(z) dz\\ =& 1{,}770{,}125.49.\end{aligned}$$

☐

9.4 A MULTISUPPLIER MODEL

In this section, we discuss a model by Dada et al. (2007) in which a newsvendor orders from multiple suppliers, some of which may be subject to supply uncertainty. The newsvendor's objective is to maximize its expected profit. Supply uncertainty is modeled in a very general way; disruptions, yield uncertainty, and many other forms of supply uncertainty are special cases. The model can be used to make ordering decisions, but also to answer qualitative questions about which suppliers will be preferred over which others.

This is a single-period model. Therefore, the ordering decision is used to balance among the suppliers' uncertainties—that is, to choose a portfolio of suppliers and corresponding order quantities in order to achieve as close as possible to the desired total inventory for the current period. In contrast, the disruption and yield uncertainty models in Sections 9.2 and 9.3 use the ordering decision to obtain inventory that can be used to buffer against the uncertainty in future orders. In other words, the earlier models spread the supply risk temporally across orders, whereas the model in this section spreads the supply risk spatially across suppliers.

9.4.1 Problem Statement

We consider a newsvendor that faces stochastic demand represented by a random variable D with pdf $f(\cdot)$ and cdf $F(\cdot)$. There are N suppliers available to the newsvendor, which may differ in terms of their supply uncertainty. If the newsvendor orders a quantity Q_i from supplier i, then the number of units supplier i will actually have available for the newsvendor is given by its *production function*, $B_i(Q_i, X_i)$, where X_i is a nonnegative random variable with pdf $g_i(\cdot)$ and cdf $G_i(\cdot)$. The suppliers never supply more than the newsvendor orders. The actual amount delivered to the newsvendor, then, is $W_i(Q_i, X_i) \equiv \min\{Q_i, B_i(Q_i, X_i)\}$. We will often drop the arguments and just write or B_i or W_i.

We assume the X_i are independent. We also assume that the $B_i(\cdot)$ are differentiable and that $\partial B_i/\partial Q_i \leq 1$ (so ordering one additional unit results in no more than one additional unit being available).

In some cases, the production function depends on the order quantity—for example, in the case of additive or multiplicative yield. We call these *endogenous production functions*. In contrast, *exogenous production functions* do not depend on the order quantity—disruptions are an example. If supplier i has an exogenous production function, then $B_i(Q_i, X_i) = X_i$ for all Q_i and X_i.

We say that supplier i is *perfectly reliable* if there is no chance that the supplier will deliver less than the newsvendor ordered from it, and *unreliable* if there is some possibility that the supplier will deliver less than ordered. In other words, i is perfectly reliable if $\mathbb{P}(W_i(Q_i, X_i) = Q_i) = 1$, and unreliable if $\mathbb{P}(W_i(Q_i, X_i) = Q_i) < 1$.

The production function is a very flexible construct, which is capable of modeling many forms of supply uncertainty. For example:

- *Perfect reliability*: $B_i(Q_i, X_i) = X_i$ and $X_i = \infty$, regardless of Q_i. In this case, we have $W_i = Q_i$.

- *Disruptions*: $B_i(Q_i, X_i) = X_i$, where $X_i = M$ with some probability and $X_i = 0$ with 1 minus that probability, for large M.

- *Capacity uncertainty*: $B_i(Q_i, X_i) = X_i$, where X_i is a random variable with a given distribution.

- *Multiplicative yield uncertainty*: $B_i(Q_i, X_i) = Q_i X_i$, where X_i is a random variable with a given distribution with support in $[0, 1]$.

- *Additive yield uncertainty*: $B_i(Q_i, X_i) = (Q_i - X_i)^+$, where X_i is a nonnegative random variable with a given distribution. (The $(\cdot)^+$ is required to ensure that the number of available units is nonnegative.)

Note that the first three examples use exogenous production functions (in fact, the first example is a special case in which X_i is degenerate, i.e., the supply is deterministic), while the last two use endogenous functions.

The newsvendor seeks to maximize its expected profit. (Most of the other models in this and earlier chapters assume the inventory manager wishes to minimize the expected cost, but the two are mathematically equivalent.) We will model the cost and revenue parameters explicitly, similar to the "explicit" newsvendor formulation in Section 4.3.2.4.

Each unit that supplier i delivers to the newsvendor costs the newsvendor c_i. (Note that this cost is charged based on the number of units delivered, not the number ordered.) We assume the suppliers are sorted so that

$$c_1 \le c_2 \le \cdots \le c_N. \tag{9.34}$$

The newsvendor earns a revenue of r for each unit that it sells. Unmet demands incur a stockout cost of p in addition to the lost profit (e.g., p is a loss-of-goodwill cost). Excess inventory may be salvaged to earn a revenue of v (with $v \le c_1$).

9.4.2 Expected Profit

Suppose we order only from supplier i. Then the underage cost per unit of unmet demand is $p + r - c_i$ and the overage cost per unit of excess inventory is $c_i - v$. The critical ratio is therefore $(p + r - c_i)/(p + r - v)$, which we denote α_i. Note that this is identical to the critical ratio in (4.21) for the explicit formulation of the classical newsvendor problem, except that there is no additional holding cost h. By (9.34), we have

$$\alpha_1 \ge \alpha_2 \ge \cdots \ge \alpha_N. \tag{9.35}$$

Let $\mathbf{Q}$ be the vector of order quantities. Then the newsvendor's expected profit as a function of $\mathbf{Q}$, denoted $\Pi(\mathbf{Q})$, is

$$\Pi(\mathbf{Q}) = \mathbb{E}_{\mathbf{X},D}\left[r \min\{D, W_T\} + v(W_T - D)^+ - p(D - W_T)^+ - \sum_{i=1}^{N} c_i W_i\right], \tag{9.36}$$

where $\mathbf{X}$ is the vector of random variables X_i and $W_T = \sum_{i=1}^N W_i$ is the total quantity delivered by all suppliers. Note that although $\mathbf{X}$ does not appear explicitly inside the expectation in (9.36), it is still there, since W_i and W_T depend on it.

One can show (see Problem 9.18) that

$$\begin{aligned}\Pi(\mathbf{Q}) =& (r + p - v)\mathbb{E}_{\mathbf{X},D}\left[\sum_{i=1}^{N} \alpha_i W_i - (W_T - D)^+\right] - p\mathbb{E}[D] & (9.37)\\ =& (r + p - v)\left[\sum_{i=1}^{N} \alpha_i \mathbb{E}_{X_i}[W_i] - \mathbb{E}_{\mathbf{X},D}[(W_T - D)^+]\right] - p\mathbb{E}[D]. & (9.38)\end{aligned}$$

The terms $r + p - v$ and $p\mathbb{E}[D]$ are constants and can be ignored without changing the function's optimizers. Therefore, we will work with the following modified expected profit function, denoted with a tilde:

$$\tilde{\Pi}(\mathbf{Q}) = \sum_{i=1}^{N} \alpha_i \mathbb{E}_{X_i}[W_i] - \mathbb{E}_{\mathbf{X},D}[(W_T - D)^+]. \tag{9.39}$$

Below, it will be convenient to separate out the parts of $\tilde{\Pi}(\mathbf{Q})$ that correspond to a given supplier i from those that do not. To that end, let $W_{T-i} \equiv W_T - W_i$ be the total quantity delivered by all suppliers except i. Then for any i,

$$\tilde{\Pi}(\mathbf{Q}) = \alpha_i \mathbb{E}_{X_i}[W_i] - \mathbb{E}_{\mathbf{X},D}[(W_i + W_{T-i} - D)^+] + \mathbb{E}_{\mathbf{X}}\left[\sum_{j \neq i}^{N} \alpha_j W_j\right], \tag{9.40}$$

where the notation $\sum_{j\neq i}^{N}$ means the sum over all $j = 1, \ldots, N$ excluding $j = i$. One can interpret (9.40) as something analogous to a standard newsvendor objective function with a single ordering decision, W_i, by treating the newsvendor's demand as $W_{T-i} - D$ (which, in (9.40), is the remaining demand after the other suppliers' delivered units are used up). The third term is a constant (with respect to W_i). If one multiplies (9.40) by -1 to convert it to a cost function, it takes the form of (4.12).

Therefore, we will consider the following optimization problem:

$$\text{maximize} \quad \tilde{\Pi}(\mathbf{Q}) \tag{9.41}$$

$$\text{subject to} \quad Q_j \geq 0 \qquad \forall j = 1, \ldots, N \tag{9.42}$$

In the objective function (9.41), $\tilde{\Pi}(\mathbf{Q})$ can be written as in (9.39) or (9.40).

9.4.3 Optimality Conditions

The objective function $\tilde{\Pi}(\mathbf{Q})$ is not, in general, concave. This can make it difficult to find the optimal order quantities. Nevertheless, the model itself still provides plenty of structure to enable us to draw interesting conclusions and insights.

We can rewrite the objective function using $\bar{n}(\cdot)$, the complementary loss function corresponding to the demand distribution (see (4.5)):

$$\tilde{\Pi}(\mathbf{Q}) = \alpha_i \mathbb{E}_{X_i}[W_i] - \mathbb{E}_{\mathbf{X}}[\bar{n}(W_i + W_{T-i})] + \mathbb{E}_{\mathbf{X}}\left[\sum_{j \neq i}^{N} \alpha_j W_j\right]. \tag{9.43}$$

Under some fairly mild conditions on the random variables and the production functions, $\tilde{\Pi}(\mathbf{Q})$ is differentiable. (See Dada et al. (2007) for details.) Let

$$\tilde{\Pi}_i'(\mathbf{Q}) \equiv \frac{\partial \tilde{\Pi}(\mathbf{Q})}{\partial Q_i}.$$

(Throughout, we will use the prime symbol $'$ to denote (partial) differentiation with respect to Q_i.) Then from (9.43), we get the following optimality condition:

$$\tilde{\Pi}_i'(\mathbf{Q}) = \alpha_i \mathbb{E}_{X_i}[W_i'(Q_i, X_i)] - \mathbb{E}_{\mathbf{X}}[W_i'(Q_i, X_i) F(W_i + W_{T-i})], \tag{9.44}$$

where $W_i'(Q_i, X_i)$ is the marginal quantity delivered by supplier i, that is,

$$W_i'(Q_i, X_i) = \begin{cases} B_i', & \text{if } B_i < Q_i \\ 1, & \text{if } B_i \geq Q_i. \end{cases} \tag{9.45}$$

(W_i is not differentiable at $B_i = Q_i$, but we will write W_i' nevertheless.) In (9.44), we use the derivative of $\bar{n}$ from (C.16), the chain rule, and the fact that the third term of (9.43) is independent of Q_i.

☐ **EXAMPLE 9.8**

Consider the special case in which there is only one supplier and it is perfectly reliable—in other words, we have a classical newsvendor. Then (dropping the subscripts i) $B(Q, X) = \infty$, $W = \min\{Q, B(Q, X)\} = Q$, $W'(Q, X) = 1$ for all Q and X, and

$$\tilde{\Pi}'(Q) = \alpha - F(Q),$$

which yields the familiar newsvendor optimality condition from Section 4.3.2.3.

Now consider the additive yield uncertainty model from Section 9.3.2.2: There is a single supplier with additive yield uncertainty, and the demand is deterministic and equal to d. We'll assume the yield is always less than or equal to Q, and that the yield is negative with very low probability; that is, $B(Q, X) = Q - X$, where X is nonnegative and $\mathbb{P}(X > Q) \approx 0$. Then $W = \min\{Q, B(Q, X)\} = B(Q, X)$, and

$$W'(Q, X) = B'(Q, X) = 1.$$

Then

$$\tilde{\Pi}'(Q) = \alpha - \mathbb{E}_X[F(Q - X)].$$

However, the demand distribution is degenerate: $F(z)$ equals 1 if $z \geq d$ and 0 otherwise. Therefore,

$$\tilde{\Pi}'(Q) = \alpha - \mathbb{E}_X[1\{Q - X \geq d\}],$$

where $1\{\cdot\} = 1$ if the condition inside the $\{\cdot\}$ is true and 0 otherwise. So

$$\tilde{\Pi}'(Q) = \alpha - \mathbb{P}(X \leq Q - d) = \alpha - G(Q - d),$$

which means that

$$Q^* = d + G^{-1}(\alpha). \tag{9.46}$$

This is identical to the optimal solution given in (9.28).[2] ☐

Any optimal solution $\mathbf{Q}^*$ satisfies the following KKT conditions:

$$\tilde{\Pi}_i'(\mathbf{Q}^*) \leq 0 \quad \forall i = 1, \ldots, N \tag{9.47}$$

$$\tilde{\Pi}_i'(\mathbf{Q}^*)Q_i^* = 0 \quad \forall i = 1, \ldots, N \tag{9.48}$$

[2]We modeled the yield as $Q - X$ ($X \geq 0$) here. In Section 9.3.2.2, we assume the yield is $Q + Y$, so to keep the models equivalent, we would assume $Y \leq 0$, with $Y = -X$. Therefore, if $F_Y(\cdot)$ is the cdf of Y, we have $G(x) = 1 - F_Y(-x)$ and $G^{-1}(\alpha) = -F_Y^{-1}(1 - \alpha)$. Thus, (9.46) and (9.28) are equivalent.

These are necessary conditions, but they are not sufficient unless $\tilde{\Pi}(\mathbf{Q})$ happens to be concave.

Suppose $Q_i^* > 0$; then by (9.48), $\tilde{\Pi}_i'(\mathbf{Q}^*) = 0$—that is,

$$\frac{\mathbb{E}_{\mathbf{X}}[W_i'(Q_i, X_i)F(W_i + W_{T-i})]}{\mathbb{E}_{X_i}[W_i'(Q_i, X_i)]} = \alpha_i. \tag{9.49}$$

In other words, we should choose Q_i so that the resulting total delivery quantity yields a service rate that, after scaling by constants that reflect supplier i's unreliability, equals the critical ratio. In the special cases in Example 9.8, these constants equal 1, but in general they need not.

9.4.4 Supplier Selection

In this section, we discuss some properties of the optimal suppliers to choose from among the available suppliers $1, \ldots, N$. The main insight from this analysis is that cost, rather than reliability, is the primary driver for supplier selection.

Let $\mathrm{SL}(\mathbf{Q})$ be the type-1 service level (see Section 4.3.4.2) resulting from the order-quantity vector $\mathbf{Q}$; i.e.,

$$\mathrm{SL}(\mathbf{Q}) = \mathbb{E}_{\mathbf{X}}[F(W_T)]. \tag{9.50}$$

Let $Q_T = \sum_{i=1}^N Q_i$. The following lemma establishes a relationship between the overall service level and the critical ratio for supplier i, based on the order quantity for supplier i.

Lemma 9.6

(a) If $Q_i^* = 0$, then $\mathrm{SL}(\mathbf{Q}^*) \ge \alpha_i$.

(b) If $Q_i^* > 0$, then $\mathbb{E}[W_i'(Q_i^*, X_i)]\alpha_i \le \mathrm{SL}(\mathbf{Q}^*) \le \alpha_i$.

Proof. Omitted; see Problem 9.19 for one part of the proof. ■

If supplier i is perfectly reliable, then Lemma 9.6 says that $\mathrm{SL}(\mathbf{Q}^*) = \alpha_i$ since $W_i'(Q_i^*, X_i) = 1$. In other words, if we order from a perfectly reliable supplier, then the overall service level is exactly equal to the service level in the classical newsvendor problem.

Theorem 9.7 *Suppose $\mathbf{Q}^*$ is an optimal solution to* (9.41)–(9.42). *Let $Q^0 \equiv F^{-1}(\alpha_1)$ be the classical newsvendor order quantity, i.e., the order quantity for a newsvendor who has only a single supplier with cost c_1 that is perfectly reliable. Then*

$$\mathrm{SL}(\mathbf{Q}^*) \le \alpha_1 = F(Q^0) \le F(Q_T^*).$$

Proof. We prove the theorem for the case in which all suppliers have exogenous supply functions. The analysis for endogenous supply functions is similar but more complex.

From (9.44) and the law of total expectation,

$$\begin{aligned}
\tilde{\Pi}'(\mathbf{Q}) =& \mathbb{E}_{\mathbf{X}}\left[W_i'(Q_i, X_i)(\alpha_i - F(W_i + W_{T-i})) \mid X_i < Q_i\right] \mathbb{P}(X_i < Q_i) + \\
& \mathbb{E}_{\mathbf{X}}\left[W_i'(Q_i, X_i)(\alpha_i - F(W_i + W_{T-i})) \mid X_i \ge Q_i\right] \mathbb{P}(X_i \ge Q_i) \\
=& \mathbb{E}_{\mathbf{X}}\left[\alpha_i - F(Q_i + W_{T-i}) \mid X_i \ge Q_i\right] \mathbb{P}(X_i \ge Q_i)
\end{aligned}$$

$$=(\alpha_i - \mathbb{E}_{\mathbf{X}}[F(Q_i + W_{T-i})])(1 - G_i(Q_i)). \tag{9.51}$$

The second equality follows from the fact that if supplier i has an exogenous supply function, then when $X_i < Q_i$, $W_i'(Q_i, X_i) = \partial X_i/\partial Q_i = 0$, and when $X_i \geq Q_i$, $W_i = Q_i$ and $W_i'(Q_i, X_i) = 1$. The third follows from the fact that W_{T-i} is independent of X_i.

One can show that $\tilde{\Pi}_i'(\mathbf{0}) > 0$; therefore, it is not optimal to set $\mathbf{Q} = \mathbf{0}$ since this violates the first KKT condition (9.47). Therefore, by Lemma 9.6(b), there exists an i such that $\mathrm{SL}(\mathbf{Q}^*) \leq \alpha_i$. By the sorting of the suppliers (9.35), $\alpha_i \leq \alpha_1$ for all i. Therefore, $\mathrm{SL}(\mathbf{Q}^*) \leq \alpha_1$, proving the first inequality. The equality comes from the definition of Q^0.

To prove the second inequality, note that

$$\alpha_1 \leq \mathbb{E}_{\mathbf{X}}[F(Q_1^* + W_{T-1}^*)]$$

by (9.47) and (9.51), where W_{T-1}^* is W_{T-1} under $\mathbf{Q} = \mathbf{Q}^*$. Then

$$\alpha_1 \leq \mathbb{E}_{\mathbf{X}}[F(Q_1^* + Q_{T-1}^*)] = \mathbb{E}[F(Q_T^*)],$$

where $Q_{T-1}^* = Q_T^* - Q_1^*$. The inequality follows from the fact that $S_i^* \leq Q_i^*$ for all i and that $F(\cdot)$ is a nondecreasing function. ■

Theorem 9.7 says that the newsvendor with unreliable suppliers orders more than the classical newsvendor (because $F(Q^0) \leq F(Q_T^*)$ and $F(\cdot)$ is increasing) but provides worse service to its customers.

We say that supplier i is *active* if $Q_i^* > 0$ and *inactive* otherwise. The next theorem gives conditions under which we know for sure that a given supplier will be inactive, given the status of lower-cost suppliers.

Theorem 9.8 *Suppose $i < j$.*

(a) If $Q_i^* = 0$, then $Q_j^* = 0$.

(b) If i is perfectly reliable, then $Q_j^* = 0$.

(c) If i is unreliable and $\mathbb{E}[W_i'(Q_i^*, X_i)]\alpha_i \geq \alpha_j$, then $Q_j^* = 0$.

Proof.

(a) If $Q_i^* = 0$, then by Lemma 9.6(a), $\mathrm{SL}(\mathbf{Q}^*) \geq \alpha_i$, and by (9.35), $\alpha_i > \alpha_j$. Therefore, $Q_j^* = 0$ by Lemma 9.6(b).

(b) If $Q_i^* = 0$, then $Q_j^* = 0$ by part (a). If, instead, $Q_i^* > 0$, then $\mathrm{SL}(\mathbf{Q}^*) = \alpha_i$ by Lemma 9.6(b); and $\alpha_i > \alpha_j$ by (9.35). Therefore, $Q_j^* = 0$ by Lemma 9.6(b).

(c) If $Q_i^* = 0$, then $Q_j^* = 0$ by part (a). If $Q_i^* > 0$, then

$$\mathrm{SL}(\mathbf{Q}^*) > \mathbb{E}[W_i'(Q_i^*, X_i)]\alpha_i \geq \alpha_j,$$

where the first inequality follows from Lemma 9.6(b) and the second is by assumption. Therefore, $Q_j^* = 0$ by Lemma 9.6(b).

■

Part (a) says that if a given supplier is inactive, then all more expensive suppliers are inactive as well. In other words, cost, not reliability, is the primary driver when choosing suppliers, since in the optimal solution, the n least expensive suppliers will be active (for some n), and no others. On the other hand, reliability is not completely irrelevant, since, according to part (b), if there is a perfectly reliable supplier i available, then all more expensive suppliers are inactive, whether i is active or not. $\mathbb{E}[W_i'(Q_i^*, X_i)]$ is a proxy for reliability: If $\mathbb{E}[W_i'(Q_i^*, X_i)] = 1$, then i is perfectly reliable, since every additional unit ordered results in exactly 1 additional unit received; and the smaller $\mathbb{E}[W_i'(Q_i^*, X_i)]$ is, the less inventory we receive for each additional unit ordered from supplier i. Therefore, part (c) gives us a hybrid measure of a supplier's quality: If $\mathbb{E}[W_i'(Q_i^*, X_i)]$ is close to 1, then supplier i is fairly reliable, and if α_i is close to 1, then supplier i is fairly inexpensive.

Taken together, the three parts of the theorem say that we activate suppliers in order of cost until we activate a supplier i either that is perfectly reliable or that has a sufficiently good combination of reliability and cost. Once we find such a supplier, it becomes active but all more expensive suppliers are inactive.

By Lemma 9.6(b), if supplier i is perfectly reliable and active, then the overall service level equals α_i, the optimal service level from the newsvendor problem in which supplier i is the sole, perfectly reliable, supplier. In other words, if there is an active, perfectly reliable supplier, then it is that supplier's role to make up the difference in the service level provided by the unreliable suppliers, bringing it up to α_i. If the unreliable suppliers' costs or reliabilities changed, their respective optimal order quantities would change, and Q_i^* would adjust to maintain a service level of α_i.

Suppose there is an expensive, unreliable supplier j that is inactive because there is a cheaper supplier i that satisfies the conditions in Theorem 9.8(b) or (c), i.e., that is perfectly reliable or for which $\mathbb{E}[W'(Q_i^*, X_i)]\alpha_i \geq \alpha_j$. Then another implication of these results is that supplier j cannot gain activation by making itself more reliable, because doing so will not change the fact that supplier i that satisfies the conditions in Theorem 9.8(b) or (c), but it *can* gain activation by making itself less expensive (thus changing the sort order so that it is preferred over i).

☐ **EXAMPLE 9.9**

A florist is preparing for Valentine's Day, when it will sell a large number of specialty roses. From previous years, the florist knows that the demand for roses on Valentine's Day will have a normal distribution with mean 500 and standard deviation 100. The florist has six suppliers of roses, which have the following characteristics:

- Supplier 1 charges the florist \$1.28 per rose. It is subject to additive yield uncertainty with $B_1(Q_1, X_1) = (Q_1 - X_1)^+$, where $X_1 \sim \exp(0.02)$.
- Supplier 2 charges \$1.46 per rose. It is subject to multiplicative yield uncertainty with $B_2(Q_2, X_2) = Q_2 X_2$, where $X_2 \sim U[0.7, 1]$.
- Supplier 3 charges \$1.49 per rose and is also subject to multiplicative yield uncertainty, with $X_3 \sim \text{Beta}(25, 2)$.
- Supplier 4 charges \$2.59 per rose. It is subject to disruptions with probability 0.2 and has infinite capacity when not disrupted. That is, $B_4(Q_4, X_4) = X_4$, where $X_4 = M$ with probability 0.8 and $X_4 = 0$ with probability 0.2. (M is a very large number.)
- Supplier 5 charges \$2.61 per rose and is perfectly reliable: $B_5(Q_5, X_5) = \infty$.

Table 9.2 Key quantities for suppliers in Example 9.9.

i	α_i	W_i	Distribution of X_i	W_i'
1	0.9628	$(Q_1 - X_1)^+$	$\exp(0.02)$	1 if $Q_1 \geq X_1$, 0 otherwise
2	0.9502	Q_2X_2	$U[0.7, 1]$	X_2
3	0.9481	Q_3X_3	Beta$(25, 2)$	X_3
4	0.8709	Q_4X_4	M w.p. 0.8, 0 w.p. 0.2	1 if $X_4 = M$, 0 otherwise
5	0.8695	Q_5	N/A	1
6	0.8667	$\min\{Q_6, X_6\}$	$N(150, 20^2)$	1 if $Q_6 \leq X_6$, 0 otherwise

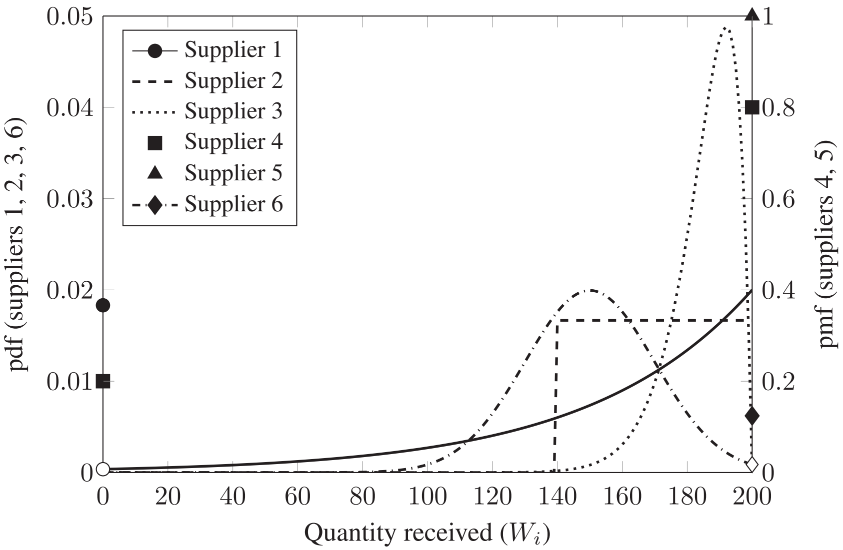


Figure 9.5 pdf/pmf of quantity received from each supplier in Example 9.9. Assumes $Q_i = 200$ for all i. Suppliers 1, 2, 3, and 6 are plotted against the left-hand y-axis; suppliers 4 and 5 are plotted against the right-hand y-axis.

- Supplier 6 charges \$2.65 per rose. It is subject to capacity uncertainty, with $B_6(Q_6, X_6) = X_6$, where $X_6 \sim N(150, 20^2)$.

The florist sells the roses for $r = \$3$ each. Unsold roses can be sold in the days after Valentine's Day for the steeply discounted price of $v = \$0.75$. Unmet demands incur a loss-of-goodwill cost of $p = \$12$ per rose.

The key quantities for each supplier are listed in Table 9.2. Figure 9.5 plots the pdf/pmf of the quantity received from each supplier if we order 200 units from each.

The supplier's modified expected profit function, from (9.39), is:

$$\begin{aligned}\tilde{\Pi}(\mathbf{Q}) =& \mathbb{E}_{\mathbf{X}}\left[\alpha_1(Q_1 - X_1)^+ + \alpha_2 Q_2 X_2 + \alpha_3 Q_3 X_3 + \alpha_4 Q_4 X_4 + \alpha_5 Q_5\right. \\ & \left. + \alpha_6 \min\{Q_6, X_6\}\right] \\ & - \mathbb{E}_{\mathbf{X},D}\left[\Big((Q_1 - X_1)^+ + Q_2X_2 + Q_3X_3 + Q_4X_4 + Q_5\right. \\ & \left. + \min\{Q_6, X_6\} - D\Big)^+\right] \end{aligned} \tag{9.52}$$

Since supplier 5 is perfectly reliable, by Theorem 9.8(b), $Q_6^* = 0$. Moreover,

$$\mathbb{E}[W_3'(Q_3^*, X_3)]\alpha_3 = \mathbb{E}[X_3]\alpha_3 = \frac{25}{27} \cdot 0.9481 = 0.8778,$$

since the mean of a Beta(α, β) random variable is $\alpha/(\alpha + \beta)$. Therefore,

$$\mathbb{E}[W_3'(Q_3^*, X_3)]\alpha_3 > \alpha_4 = 0.8709,$$

so by Theorem 9.8(c), $Q_4^* = 0$. Finally, by Theorem 9.8(a), $Q_5^* = 0$. Therefore, (9.52) simplifies to

$$\begin{aligned}\tilde{\Pi}(\mathbf{Q}) =& \mathbb{E}_{\mathbf{X}}\left[\alpha_1(Q_1 - X_1)^+ + \alpha_2 Q_2 X_2 + \alpha_3 Q_3 X_3\right] \\ &- \mathbb{E}_{\mathbf{X},D}\left[(Q_1 - X_1)^+ + Q_2 X_2 + Q_3 X_3\right]. \end{aligned} \tag{9.53}$$

Unfortunately, $\tilde{\Pi}(\mathbf{Q})$ is not concave. (An easy way to see this is to note that in the first line of (9.53),

$$\mathbb{E}_{\mathbf{X}}[\alpha_1(Q_1 - X_1)^+] = \alpha_1 \mathbb{E}_{X_1}[(Q_1 - X_1)^+] = \bar{n}(Q_1),$$

which is convex.) Moreover, (9.53) is computationally expensive to evaluate. Therefore, optimizing (9.53) to find $\mathbf{Q}^*$ is nontrivial. In the next example, we consider a simpler instance whose objective function is concave and can be calculated efficiently. □

□ **EXAMPLE 9.10**

Suppose that suppliers 1–3 in Example 9.9 are all subject to multiplicative yield uncertainty, with $X_i \sim N(\mu_i, \sigma_i^2)$. In particular, $(\mu_i, \sigma_i) = (0.5, 0.1)$, $(0.4, 0.05)$, and $(0.9, 0.02)$ for $i = 1, 2, 3$, respectively. Note that for all three suppliers, the probability that X_i is outside $[0, 1]$ is negligible. The pdf of the quantity received for these suppliers, assuming we order 200 units, is plotted in Figure 9.6.

We have

$$\mathbb{E}[W_3'(Q_3^*, X_3)]\alpha_3 = \mathbb{E}[X_3]\alpha_3 = 0.92 \cdot 0.9481 = 0.8722 > 0.8709 = \alpha_4,$$

so by Theorem 9.8, $Q_4^* = Q_5^* = Q_6^* = 0$. The modified expected profit function is therefore

$$\tilde{\Pi}(\mathbf{Q}) = \mathbb{E}_{\mathbf{X}}\left[\sum_{i=1}^{3} \alpha_i Q_i X_i\right] - \mathbb{E}_{\mathbf{X},D}\left[\left(\sum_{i=1}^{3} Q_i X_i - D\right)^+\right].$$

Let

$$Y = \sum_{i=1}^{3} Q_i X_i - D.$$

Since D and each of the R_i are normally distributed, so is Y; it has a mean of $\sum_{i=1}^{3} \mu_i Q_i - 500$ and a variance of $\sum_{i=1}^{3} \sigma_i^2 Q_i + 100$. (Recall that $D \sim N(500, 100^2)$.) Then we have

$$\tilde{\Pi}(\mathbf{Q}) = \sum_{i=1}^{3} \alpha_i \mu_i Q_i - \mathbb{E}[Y^+] = \sum_{i=1}^{3} \alpha_i \mu_i Q_i - n_Y(0), \tag{9.54}$$

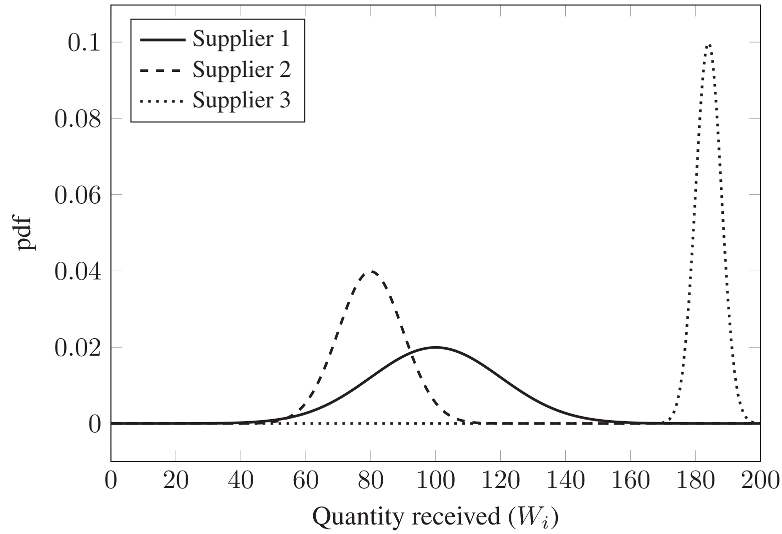


Figure 9.6 pdf of quantity received from each supplier in Example 9.10. Assumes $Q_i = 200$ for all i.

where $n_Y(\cdot)$ is the loss function for Y. The expression in (9.54) is concave (because the first term is linear and $n_Y(\cdot)$ is convex), and it can be evaluated efficiently. Figure 9.7 plots $\tilde{\Pi}(\mathbf{Q})$ as Q_1 and Q_2 vary, keeping Q_3 fixed at 100.

One can optimize (9.54) using a convex optimization solver (we used MATLAB's `fmincon` function). The resulting optimal solution is

$$\mathbf{Q}^* = (988.48, 455.45, 60.04)$$

with objective function $\tilde{\Pi}(\mathbf{Q}^*) = 466.56$. Let $F_Y(\cdot)$ be the cdf of Y; then the service level $SL(\mathbf{Q}^*)$ attained by this solution is

$$P\left(\sum Q_i X_i - D \geq 0\right) = \mathbb{P}(Y \geq 0) = 1 - F_Y(0) = 0.9480.$$

Therefore, $\text{SL}(\mathbf{Q}^*) \geq \alpha_i$ and $Q_i^* = 0$ for $i = 4, 5, 6$, confirming Lemma 9.6(a); and

$$\begin{aligned}
\mathbb{E}[W_1'(Q_1^*, X_1)]\alpha_1 &= 0.50 \cdot 0.9628 = 0.4814 \leq \text{SL}(\mathbf{Q}^*) \leq 0.9628 = \alpha_1 \\
\mathbb{E}[W_2'(Q_2^*, X_2)]\alpha_2 &= 0.40 \cdot 0.9502 = 0.3801 \leq \text{SL}(\mathbf{Q}^*) \leq 0.9502 = \alpha_2 \\
\mathbb{E}[W_3'(Q_3^*, X_3)]\alpha_3 &= 0.92 \cdot 0.9481 = 0.8722 \leq \text{SL}(\mathbf{Q}^*) \leq 0.9481 = \alpha_3,
\end{aligned}$$

confirming Lemma 9.6(b). We have $F(Q_T^*) = F(1503.96) \approx 1.00$, so if all Q_T^* units ordered were actually delivered, there would be virtually no chance of a stockout. In other words,

$$\text{SL}(\mathbf{Q}^*) \leq 0.9628 = \alpha_1 = F(Q^0) \leq 1.0000 = F(Q_T^*),$$

confirming Theorem 9.7.

Finally, suppose that supplier 3 was perfectly reliable, i.e., that $\mu_3 = 1$ and $\sigma_3 = 0$. Then the optimal order quantities are

$$\mathbf{Q}^* = (985.96, 450.71, 58.06)$$

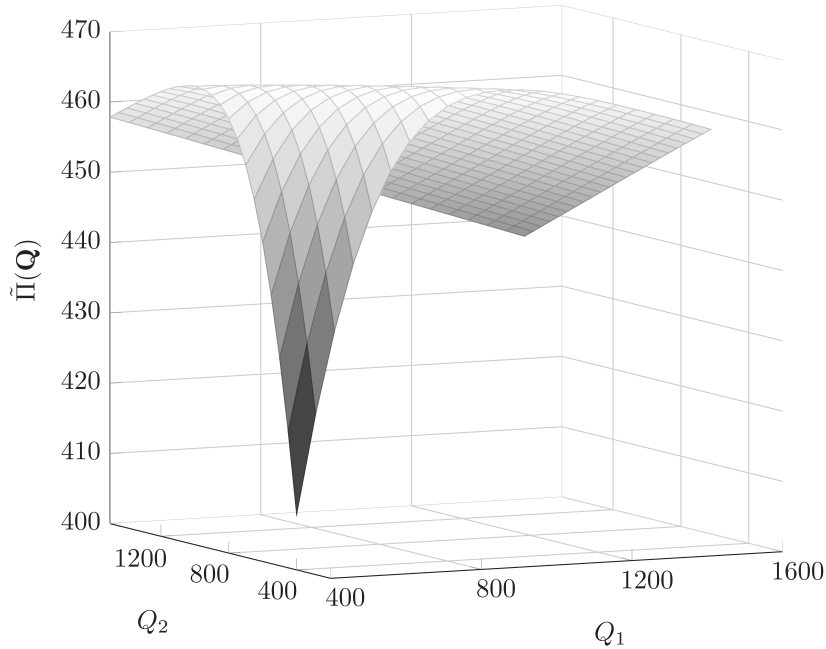


Figure 9.7 $\tilde{\Pi}(\mathbf{Q})$ for Example 9.10, varying Q_1 and Q_2 while keeping Q_3 fixed at 100.

and the corresponding service level is $SL(\mathbf{Q}^*) = 0.9481 = \alpha_3$. A small number of items shift from suppliers 1 and 2 to supplier 3 to take advantage of its perfect reliability and, as expected from Lemma 9.6(b), the overall service level equals the service level that would be attained in the classical newsvendor problem with supplier 3 as the only supplier. □

9.4.5 Closing Thoughts

The results discussed above suggest that if a newsvendor's suppliers differ in terms of both cost and reliability, cost generally takes precedence over reliability when selecting suppliers. We rank the suppliers in terms of cost, and then start placing orders, supplier by supplier. As the total order quantity increases, so does $SL(\mathbf{Q}^*)$. We continue adding suppliers until we reach a supplier j that is either perfectly reliable (by Theorem 9.8) or for which $\alpha_{j+1} \le SL(\mathbf{Q}^*) \le \alpha_j$ (by Lemma 9.6). The actual order quantities from each supplier, $Q_1^*, \ldots, Q_j^*$, must be found using numerical optimization, except in special cases. The overall service level attained by the optimal solution is no greater than the service level that would be attained if the newsvendor had only a single, perfectly reliable supplier, with supplier 1's costs (by Theorem 9.7).

This analysis suggests that an expensive supplier cannot gain activation (i.e., cannot convince the newsvendor to give it a nonzero order quantity) by improving its reliability, because reliability does not affect the ranking by α_i. An expensive supplier can only gain activation by improving its cost. In contrast, an unreliable supplier can gain activation by improving its cost, even if it remains unreliable. Once a supplier is active, however, it can increase its share of the total order by improving its cost and/or reliability.

9.5 THE RISK-DIVERSIFICATION EFFECT

9.5.1 Problem Statement

Consider the N-DC system described in Section 7.2, except that now the demand is deterministic and equal to d per period ($\mu_i = d$, $\sigma_i^2 = 0$ for all i) but the supply may be disrupted. All DCs follow a periodic-review base-stock policy, as in Section 9.2.2. Disruptions follow the same two-state Markov process described in Section 9.2.2, with disruption probability α and recovery probability β. As before, π_n is the pmf of the disruption process and $F(n)$ is the cdf.

The central question is, would it be preferable to consolidate the N DCs into a single DC? That is, is a centralized system preferable to a decentralized one? It turns out that the decentralized system is preferable in this case, but not because it has a lower expected cost. In fact, the two systems have the *same* expected cost, but the decentralized system has a lower variance. Therefore, risk-averse decision makers would prefer the decentralized system.

This phenomenon—whereby the cost variance (but not the mean cost) is smaller when inventory is held at a decentralized set of locations—is called the *risk-diversification effect*. Intuitively, it occurs because a given DC (or its portion of the central DC) is disrupted the same number of times, on average, in both systems, but disruptions are more severe in the centralized system. The supply chain benefits by not having all its eggs in one basket. The risk-diversification effect was first described by Snyder and Shen (2006), who demonstrated it using simulation; the theoretical analysis in this section is based on Schmitt et al. (2015).

Note the parallels to the risk-pooling effect: Whereas the risk-pooling effect says that the mean cost (but not the variance (Schmitt et al. 2015)) is lower in a centralized system under demand uncertainty, the risk-diversification effect says that the cost variance is lower (and the mean cost is equal) in a decentralized system under supply uncertainty.

In fact, Snyder and Shen (2006) comment that supply uncertainty (in the form of disruptions) often has a mirror-image effect in relation to demand uncertainty, and that the optimal strategy under one type of uncertainty is often the exact opposite of that under the other type of uncertainty. The risk-diversification effect is an example of this mirror-image phenomenon, in the sense that supply chains under supply uncertainty behave in the opposite way to the ways we've observed them behaving previously, under demand uncertainty.

9.5.2 Notation

Let

$$\hat{g}(S, n) = h\left[S - (n+1)d\right]^+ + p\left[(n+1)d - S\right]^+$$

be the cost in a given period in a single-stage system if we use a base-stock level of S and are in the nth period of a disruption ($n \geq 0$). Then from (9.14),

$$g(S) = \sum_{n=0}^{\infty} \pi_n \hat{g}(S, n).$$

Let $g^* = g(S^*)$ and V^* be the mean and variance of the optimal cost:

$$g^* = \mathbb{E}\left[\hat{g}(S^*, n)\right] = g(S^*)$$

$$V^* = \mathrm{Var}\left[\hat{g}(S^*, n)\right],$$

where the expectation and variance are taken over the disruption state, n. We'll use subscripts D and C to refer to the costs in the decentralized and centralized systems, respectively, and no subscript when we're discussing a single-stage system. Asterisks denote optimal solutions.

9.5.3 Optimal Solution

The optimal base-stock level for a single-stage newsvendor system with disruptions is given by Theorem 9.5:

$$S^* = d + dF^{-1}\left(\frac{p}{p+h}\right). \tag{9.55}$$

(Remember that $F^{-1}(\gamma)$ is interpreted as the smallest n such that $F(n) \geq \gamma$.)

Now, in the decentralized system, each DC acts like a single-stage system, so the optimal base-stock level at each DC is $S_D^* = S^*$, where S^* is given by (9.55). In the centralized system, the warehouse acts as a single stage facing a demand of Nd. Therefore, its optimal base-stock level is

$$S_C^* = Nd + NdF^{-1}\left(\frac{p}{p+h}\right) = NS_D^* = NS^*.$$

Thus, the total inventory is the same in both the centralized and decentralized systems. (In contrast, the total inventory is smaller in the centralized system under the risk-pooling effect, assuming $h < p$.)

9.5.4 Mean and Variance of Optimal Cost

Next, we examine the mean and variance of the cost when we use the optimal base-stock levels in each system. In the decentralized system, since each DC acts like a single-stage system, the total expected cost is just N times the total expected cost in a single-stage system: $g_D^* = Ng^*$. In the centralized system, the optimal cost at the warehouse is obtained by substituting NS^* in place of S and Nd in place of d in (9.14):

$$\begin{aligned}
g_C^* &= \sum_{n=0}^{\infty} \pi_n \left[h(NS^* - (n+1)Nd)^+ + p((n+1)Nd - NS^*)^+\right] \\
&= N\sum_{n=0}^{\infty} \pi_n \left[h(S^* - (n+1)d)^+ + p((n+1)d - S^*)^+\right] \\
&= Ng^* = g_D^*
\end{aligned} \tag{9.56}$$

Therefore, the expected cost is the same in the centralized and decentralized systems when we set the base-stock levels optimally in each. In both systems, each DC experiences disruption-related stockouts in the same percentage of periods. Moreover, during nondisrupted periods, the two systems have the same amount of inventory. Therefore, the optimal expected cost is the same in both systems.

Rather than improving the mean cost, decentralization improves the cost variance. Intuitively, this is because disruptions in the centralized system are less frequent but more severe, and therefore, they cause greater variability. To prove this mathematically, first note that

$$V_D^* = NV^* \tag{9.57}$$

because the decentralized system consists of N individual single-stage systems. Recall that $\mathrm{Var}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$ and note that, for a single-stage system,

$$\mathbb{E}\left[\hat{g}(S^*, n)^2\right] = \sum_{n=0}^{\infty} \pi_n \left[h^2 \left((S^* - (n+1)d)^+\right)^2 + p^2 \left(((n+1)d - S^*)^+\right)^2\right]. \tag{9.58}$$

Similarly, in the centralized system,

$$\begin{aligned}
\mathbb{E}[\hat{g}_C(S_C^*, n)^2] &= \sum_{n=0}^{\infty} \pi_n \left[h^2 \left((NS^* - (n+1)Nd)^+\right)^2 + p^2 \left(((n+1)Nd - NS^*)^+\right)^2\right] \\
&= N^2 \sum_{n=0}^{\infty} \pi_n \left[h^2 \left((S^* - (n+1)d)^+\right)^2 + p^2 \left(((n+1)d - S^*)^+\right)^2\right] \\
&= N^2 \mathbb{E}\left[\hat{g}(S^*, n)^2\right]
\end{aligned} \tag{9.59}$$

Then the variance in the centralized system is given by

$$\begin{aligned}
V_C^* &= \mathbb{E}[\hat{g}_C(S_C^*, n)^2] - (g_C^*)^2 \\
&= N^2 \mathbb{E}\left[\hat{g}(S^*, n)^2\right] - (Ng^*)^2 \text{ (by (9.59) and (9.56))} \\
&= N^2 (\mathbb{E}\left[\hat{g}(S^*, n)^2\right] - (g^*)^2) \\
&= N^2 V^* \\
&> NV^* = V_D^*
\end{aligned}$$

Therefore, the variance is smaller in the decentralized system—this is the risk-diversification effect. We summarize the preceding results in the following theorem:

Theorem 9.9 *For the decentralized N-DC system with supply disruptions and deterministic demand, and the centralized, single-DC system formed by merging the DCs:*

1. $S_C^* = NS_D^* = NS^*$
2. $g_C^* = g_D^* = Ng^*$
3. $V_C^* = NV_D^* = N^2 V^*$

9.5.5 Supply Disruptions and Stochastic Demand

Suppose now that demand is uncertain, as in Section 7.2. Disruptions are also still present, as in the preceding analysis.

Under demand uncertainty, the risk-pooling effect says that centralization is preferable, while under supply uncertainty, the risk-diversification effect says that decentralization is preferable. So, if both types of uncertainty are present, which strategy is better? We cannot answer this question analytically since the expected cost function cannot be optimized in closed form for either system. Instead, we evaluate the question numerically.

Most decision makers are risk averse—they are willing to sacrifice a certain amount of expected cost in order to reduce the variance of the cost. One way of modeling risk aversion is using a *mean–variance objective*, popularized by Markowitz in the 1950s:

$$(1-\kappa)g^* + \kappa V^*, \tag{9.60}$$

where $\kappa \in [0,1]$ is a constant. If κ is small, then the decision maker is fairly risk neutral; the larger κ is, the more risk-averse the decision maker is. Typically κ is less than, say, 0.05.

One can write out g^* and V^* for the systems with disruptions and demand uncertainty, but we omit the formulas here. Schmitt et al. (2015) perform a computational study to determine which system is preferable to the risk-averse decision maker. They numerically optimize (9.60) for both the centralized and decentralized systems and determine which system gives the smaller optimal objective value.

They find that the decentralized system is almost always optimal, i.e., that the risk-diversification effect almost always trumps the risk-pooling effect. For example, under a given set of problem parameters, the decentralized system is optimal whenever $\kappa \geq 0.0008$ and $p/(p+h) \geq 0.5$—in other words, whenever the decision maker is even slightly risk averse and the required service level is at least 50%.

9.6 A FACILITY LOCATION MODEL WITH DISRUPTIONS

9.6.1 Introduction

The uncapacitated fixed-charge location problem (UFLP) introduced in Section 8.2 chooses facility locations and customer assignments to minimize fixed and transportation costs. The model assumes that facilities always operate as planned. However, facilities are occasionally disrupted by weather conditions, labor actions, or natural disasters. These disruptions may result in increased costs as customers previously served by these facilities must now be served by more distant ones. The model presented in this section chooses facility locations to minimize the expected cost after accounting for disruptions. We call the ability of a system to perform well even when parts of the system are disrupted the *reliability* of the system. Our goal is to choose facility locations that are both inexpensive and reliable.

Figure 9.8 shows the optimal UFLP solution for a data set consisting of the capitals of the lower 48 United States plus Washington, DC (Daskin 1995). In this solution, the fixed cost is \$348,000, and the transportation cost is \$509,000. Now suppose that the facility in Sacramento, CA, is disrupted. During the disruption, Sacramento's customers are re-routed to their nearest open facilities, in Springfield, IL, and Austin, TX (Figure 9.9). This new solution has a transportation cost of \$1,081,000, an increase of 112%.

Table 9.3 lists the *disruption costs* (the transportation cost when a site is disrupted) of the five optimal DCs, as well as their assigned demands. From the table, it is evident that the reliability of a facility can depend either on its distance from other facilities or on the demand it serves, or both. For example, Sacramento, CA, serves a relatively small portion of the total demand, but it has a large disruption cost because its nearest "backup" facilities are far away. Harrisburg, PA, also has a high disruption cost, even though it is relatively close to two good backup facilities; the high disruption cost occurs because Harrisburg

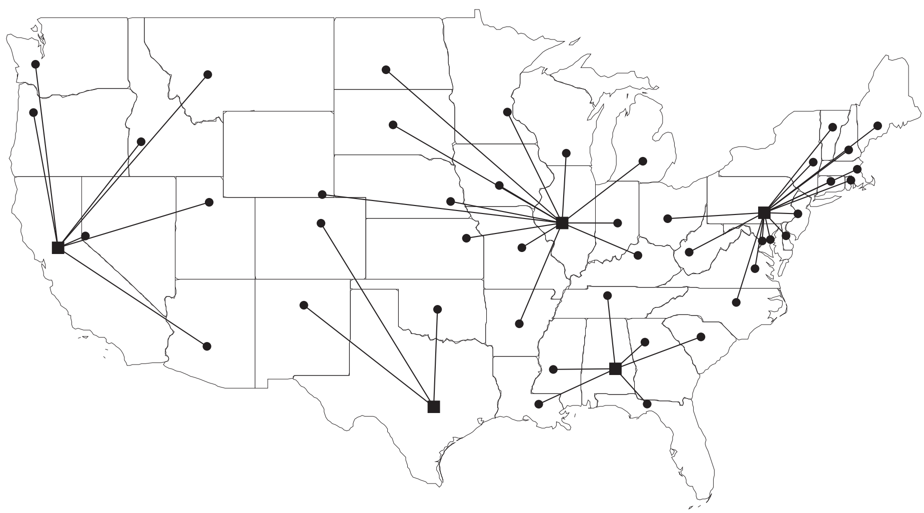


Figure 9.8 UFLP solution for 49-node data set. Reprinted by permission, Snyder and Daskin, Reliability models for facility location: The expected failure cost case, *Transportation Science*, 39(3), 2005, 400–416. ©2005, the Institute for Operations Research and the Management Sciences, 7240 Parkway Drive, Suite 300, Hanover, MD 21076 USA.

Figure 9.9 UFLP solution for 49-node data set, after disruption of facility in Sacramento. Reprinted by permission, Snyder and Daskin, Reliability models for facility location: The expected failure cost case, *Transportation Science*, 39(3), 2005, 400–416. ©2005, the Institute for Operations Research and the Management Sciences, 7240 Parkway Drive, Suite 300, Hanover, MD 21076 USA.

Table 9.3 Disruption costs for optimal DCs. Reprinted by permission, Snyder and Daskin, Reliability models for facility location: The expected failure cost case, *Transportation Science*, 39(3), 2005, 400–416. ©2005, the Institute for Operations Research and the Management Sciences, 7240 Parkway Drive, Suite 300, Hanover, MD 21076 USA.

Location	% Demand Served	Disruption Cost	% Increase
Sacramento, CA	19	1,081,229	112
Harrisburg, PA	33	917,332	80
Springfield, IL	22	696,947	37
Montgomery, AL	16	639,631	26
Austin, TX	10	636,858	25
Transportation cost w/no disruptions		508,858	0

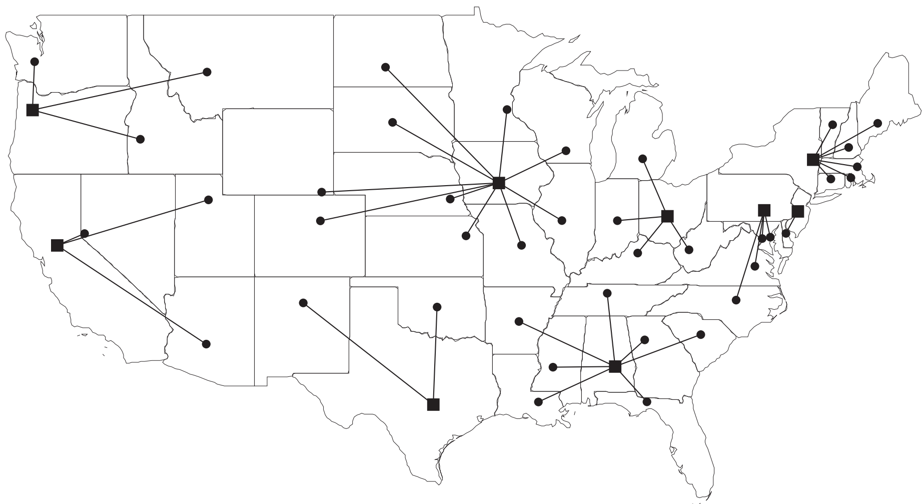


Figure 9.10 Reliable solution for 49-node data set.

serves one-third of the total demand. Springfield, IL, is the second-largest facility in terms of demand served, but its disruption cost is much smaller because it is centrally located, close to good backup facilities.

It is possible to choose facility locations that are more resilient to disruptions—that is, that have lower disruption costs. For example, suppose we locate facilities in the capitals of CA, NY, TX, PA, OH, AL, OR, and IA. (See Figure 9.10.) In this solution, every disruption cost is less than or equal to \$640,000. On the other hand, three additional facilities are used in this solution. Is the improvement in reliability worth the increased facility cost? One of the goals of the model in this section is to demonstrate that the answer is often "yes." In other words, substantial improvements in reliability can often be obtained without large increases in the UFLP cost. This means that by taking reliability into account at design time, one can find a near-optimal UFLP solution that has much better reliability.

We will present an extension of the UFLP that minimizes the expected postdisruption cost, given a certain probability that each facility is disrupted. Multiple facilities may be disrupted simultaneously. We refer to this model as the *reliable fixed-charge location* *problem* (RFLP). The model we present is a simplified version of the model introduced by Snyder and Daskin (2005). A similar model was studied by Berman et al. (2007). For reviews on facility location models with disruptions, see Snyder et al. (2006) or Snyder and Daskin (2007).

9.6.2 Notation

As in the UFLP, let I be the set of customers and J the set of potential facility sites. Let h_i be the demand at customer i, c_{ij} the transportation cost from facility j to customer i, and f_j the fixed cost to open facility j.

Each facility in J has the same probability q of being disrupted, which is interpreted as the long-run fraction of time the facility is nonoperational. In some cases, q may be estimated based on historical data (e.g., for weather-related disruptions), while in others q must be estimated subjectively (e.g., for disruptions due to labor strikes). We can assume that facility disruptions follow a two-state Markov process, as in Section 9.2, but the exact disruption process is not important. It is important, however, that disruptions are statistically independent from facility to facility.

The assumption that every facility has the same disruption probability q is generally unrealistic, but it makes the model considerably easier to solve. Several approaches have been proposed for relaxing this assumption; see, e.g., Berman et al. (2007), Li and Ouyang (2010), Cui et al. (2010), Shen et al. (2011), and also Problem 9.23.

Associated with each customer i is a cost θ_i that represents the cost of not serving the customer—for example, if all open facilities are disrupted—per unit of demand. θ_i may be a lost-sales cost, or the cost of serving i by purchasing product from a competitor on an emergency basis. Instead of modeling this eventuality explicitly, we perform a modeling trick: We add an "emergency" facility u that cannot be disrupted and we force $x_u = 1$. This facility has fixed cost $f_u = 0$ and transportation cost $c_{iu} = \theta_i$ for every customer $i \in I$. From this point forward, the set J is assumed to contain u, as well.

The strategy behind the formulation of the RFLP is to assign each customer to a primary facility that will serve it under normal circumstances, as well as to a set of backup facilities that serve it when the primary facility is disrupted. Since multiple disruptions may occur simultaneously, each customer needs a first backup facility in case its primary facility is disrupted, a second backup facility in case its first backup is disrupted, and so on.

There are two sets of decision variables in this model:

$$x_j = \begin{cases} 1, & \text{if facility } j \in J \text{ is selected} \\ 0, & \text{otherwise} \end{cases}$$

$$y_{ijr} = \begin{cases} 1, & \text{if customer } i \text{ is assigned to facility } j \text{ as a level-} r \text{ assignment} \\ 0, & \text{otherwise} \end{cases}$$

A "level-r" assignment is one for which there are r closer facilities that are open. If $r = 0$, this is a primary assignment; otherwise, it is a backup assignment. Each customer i has a level-r assignment for each $r = 0, \ldots, |J| - 1$, unless i is assigned to the emergency facility u at level s, where $s < r$. In other words, customer i is assigned to one facility at level 0, another facility at level 1, and so on until i has been assigned to facility u at some level. If a customer is assigned to facility u at a level r, with $r < |J| - 1$, then it is preferable to lose that customer's demand than to serve it from the remaining facilities if the first r facilities have failed.

9.6.3 Formulation

The objective function of the RFLP is given by

$$\sum_{j\in J} f_j x_j + \sum_{i\in I}\sum_{\substack{j\in J\\ j\neq u}}\sum_{r=0}^{|J|-1} h_i c_{ij} q^r (1-q) y_{ijr} + \sum_{i\in I}\sum_{r=0}^{|J|-1} h_i c_{iu} q^r y_{iur}.$$

This expression calculates the fixed cost plus the expected transportation cost. Each customer i is served by its level-r facility (call it j) if the r closer facilities are disrupted (this occurs with probability q^r) and if j itself is not disrupted (this occurs with probability $1-q$, unless $j = u$, in which case it occurs with probability 1). For notational convenience, we define

$$\psi_{ijr} = \begin{cases} h_i c_{ij} q^r, & \text{if } j = u \\ h_i c_{ij} q^r (1-q), & \text{if } j \neq u. \end{cases}$$

Then the RFLP can be formulated as an IP as follows:

(RFLP)

$$\text{minimize} \quad \sum_{j\in J} f_j x_j + \sum_{i\in I}\sum_{j\in J}\sum_{r=0}^{|J|-1} \psi_{ijr} y_{ijr} \tag{9.61}$$

$$\text{subject to} \quad \sum_{j\in J} y_{ijr} + \sum_{s=0}^{r-1} y_{ius} = 1 \qquad \forall i \in I, r = 0,\dots,|J|-1 \tag{9.62}$$

$$y_{ijr} \le x_j \qquad \forall i \in I, j \in J, r = 0,\dots,|J|-1 \tag{9.63}$$

$$\sum_{r=0}^{|J|-1} y_{ijr} \le 1 \qquad \forall i \in I, j \in J \tag{9.64}$$

$$x_u = 1 \tag{9.65}$$

$$x_j \in \{0,1\} \qquad \forall j \in J \tag{9.66}$$

$$y_{ijr} \in \{0,1\} \qquad \forall i \in I, j \in J, r = 0,\dots,|J|-1 \tag{9.67}$$

Constraints (9.62) require that for each customer i and each level r, either i is assigned to a level-r facility or it is assigned to facility u at a level $s < r$. (By convention we take $\sum_{s=0}^{r-1} y_{ijs} = 0$ if $r = 0$.) Constraints (9.63) prohibit an assignment to a facility that has not been opened. Constraints (9.64) prohibit a customer from being assigned to a given facility at more than one level. Constraint (9.65) requires the emergency facility u to be opened. Constraints (9.66) and (9.67) are integrality constraints.

You may be wondering why there are no constraints requiring the assignments to occur in order of distance—that is, for a customer's level-r facility to be closer than its level-$(r+1)$ facility. It turns out that this assignment strategy is always optimal, so it does not need to be enforced by constraints.

Theorem 9.10 *In any optimal solution to (RFLP), if* $y_{ijr} = y_{i,k,r+1} = 1$ *for* $i \in I$, $j, k \in J$, $0 \le r < |J| - 2$, *then* $c_{ij} \le c_{ik}$.

Proof. Omitted; see Problem 9.22. ■

9.6.4 Lagrangian Relaxation

We solve (RFLP) by relaxing constraints (9.62) using Lagrangian relaxation. For given Lagrange multipliers λ, the subproblem is as follows:

(RFLP-LR$_\lambda$)

$$\begin{aligned}
\text{minimize} \quad & \sum_{j\in J} f_j x_j + \sum_{i\in I}\sum_{j\in J}\sum_{r=0}^{|J|-1} \psi_{ijr} y_{ijr} + \\
& \qquad \sum_{i\in I}\sum_{r=0}^{|J|-1} \lambda_{ir}\left(1 - \sum_{j\in J} y_{ijr} - \sum_{s=0}^{r-1} y_{ius}\right) && (9.68)\\
\text{subject to} \quad & y_{ijr} \le x_j \qquad \forall i \in I, j \in J, r = 0, \ldots, |J|-1 && (9.69)\\
& \sum_{r=0}^{|J|-1} y_{ijr} \le 1 \qquad \forall i \in I, j \in J && (9.70)\\
& x_u = 1 && (9.71)\\
& x_j \in \{0,1\} \quad \forall j \in J && (9.72)\\
& y_{ijr} \in \{0,1\} \quad \forall i \in I, j \in J, r = 0, \ldots, |J|-1 && (9.73)
\end{aligned}$$

The portion of the objective function (9.68) other than the fixed costs can be rewritten as follows:

$$\begin{aligned}
&\sum_{i\in I}\sum_{j\in J}\sum_{r=0}^{|J|-1}(\psi_{ijr} - \lambda_{ir})y_{ijr} + \sum_{i\in I}\sum_{r=0}^{|J|-1}\lambda_{ir} - \sum_{i\in I}\sum_{r=0}^{|J|-1}\sum_{s=0}^{r-1}\lambda_{ir} y_{ius}\\
=&\sum_{i\in I}\sum_{j\in J}\sum_{r=0}^{|J|-1}(\psi_{ijr} - \lambda_{ir})y_{ijr} + \sum_{i\in I}\sum_{r=0}^{|J|-1}\lambda_{ir} - \sum_{i\in I}\sum_{s=0}^{|J|-1}\sum_{r=0}^{s-1}\lambda_{is} y_{iur}
\end{aligned}$$

(by swapping the indices r and s in the last term)

$$\begin{aligned}
=&\sum_{i\in I}\sum_{j\in J}\sum_{r=0}^{|J|-1}(\psi_{ijr} - \lambda_{ir})y_{ijr} + \sum_{i\in I}\sum_{r=0}^{|J|-1}\lambda_{ir} - \sum_{i\in I}\sum_{\substack{r=0,\ldots,|J|-1\\ s=0,\ldots,|J|-1\\ r<s}}\lambda_{is} y_{iur}\\
=&\sum_{i\in I}\sum_{j\in J}\sum_{r=0}^{|J|-1}(\psi_{ijr} - \lambda_{ir})y_{ijr} + \sum_{i\in I}\sum_{r=0}^{|J|-1}\lambda_{ir} - \sum_{i\in I}\sum_{r=0}^{|J|-1}\left(\sum_{s=r+1}^{|J|-1}\lambda_{is}\right) y_{iur}
\end{aligned}$$

Therefore, the objective function can be written as

$$\sum_{j\in J} f_j x_j + \sum_{i\in I}\sum_{j\in J}\sum_{r=0}^{|J|-1}\tilde{\psi}_{ijr} y_{ijr} + \sum_{i\in I}\sum_{r=0}^{|J|-1}\lambda_{ir}, \tag{9.74}$$

where

$$\tilde{\psi}_{ijr} = \begin{cases} \psi_{ijr} - \lambda_{ir}, & \text{if } j \neq u \\ \psi_{ijr} - \lambda_{ir} - \sum_{s=r+1}^{|J|-1} \lambda_{is} = \psi_{ijr} - \sum_{s=r}^{|J|-1} \lambda_{is}, & \text{if } j = u \end{cases} \tag{9.75}$$

For given λ, problem (RFLP-LR$_\lambda$) can be solved easily. Since the assignment constraints (9.62) have been relaxed, customer i may be assigned to zero, one, or more than one open facility at each level, but it may be assigned to a given facility at at most one level r. Suppose that facility j is opened. Customer i will be assigned to facility j at level r if $\tilde{\psi}_{ijr} < 0$ and $\tilde{\psi}_{ijr} \leq \tilde{\psi}_{ijs}$ for all $s = 0, \ldots, |J| - 1$. Therefore, the benefit of opening facility j is given by

$$\beta_j = \sum_{i \in I} \min \left\{ 0, \min_{r=0,\ldots,|J|-1} \{\tilde{\psi}_{ijr}\} \right\}. \tag{9.76}$$

Once the benefits β_j have been computed for all j, we set $x_j = 1$ for the emergency facility u and for any j for which $\beta_j + f_j < 0$; we set $y_{ijr} = 1$ if (1) facility j is open, (2) $\tilde{\psi}_{ijr} < 0$, and (3) r minimizes $\tilde{\psi}_{ijs}$ for $s = 0, \ldots, |J| - 1$. The optimal objective value for (RFLP-LR$_\lambda$) is

$$\sum_{j \in J} (\beta_j + f_j) x_j + \sum_{i \in I} \sum_{r=0}^{|J|-1} \lambda_{ir},$$

and this provides a lower bound on the optimal objective value of (RFLP).

One can obtain upper bounds by first opening the facilities that are open in the solution to (RFLP-LR$_\lambda$), then assigning customers to level-r facilities in increasing order of distance. As in the UFLP, improvement heuristics (e.g., exchange heuristics) can be applied to improve the solution found.

The Lagrange multipliers are updated using subgradient optimization in a manner very similar to that described in Section 8.2.3.5. If the procedure terminates without a provably optimal solution, branch-and-bound can be used to close the gap, as described in Section D.1.6.

9.6.5 Trade-off Curves

The RFLP can alternately be modeled as a multiobjective optimization problem in which one objective represents the normal UFLP cost (ignoring disruptions) and the other objective represents the expected transportation cost (accounting for disruptions). Multiobjective optimization allows the decision maker to express her preference between the two objectives. For example, a firm that is used to thinking only about the classical UFLP objective may weight the problem toward this objective, while a firm that is very concerned about disruptions may favor the other objective.

The two objectives can be formulated as follows:

$$w_1 = \sum_{j \in J} f_j x_j + \sum_{i \in I} \sum_{j \in J} h_i c_{ij} y_{ij0}$$

$$w_2 = \sum_{i \in I} \sum_{j \in J} \sum_{r=0}^{|J|-1} \psi_{ijr} y_{ijr}.$$

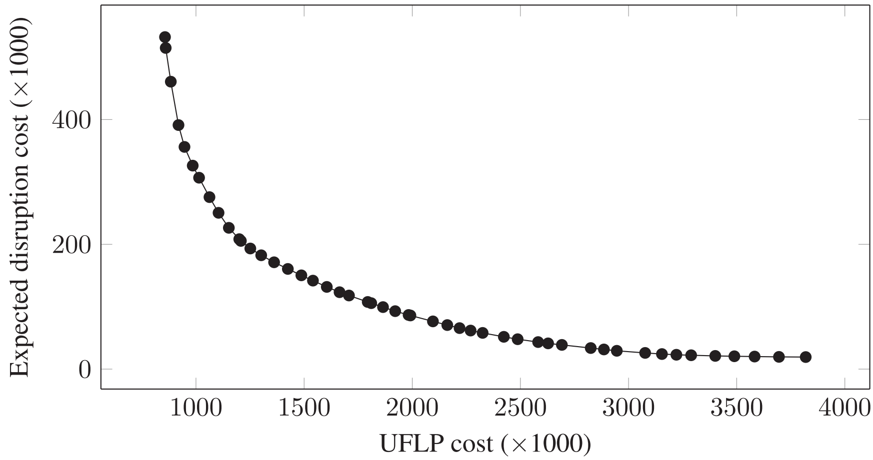


Figure 9.11 Sample RFLP trade-off curve. Reprinted by permission, Snyder and Daskin, Reliability models for facility location: The expected failure cost case, *Transportation Science*, 39(3), 2005, 400–416. ©2005, the Institute for Operations Research and the Management Sciences, 7240 Parkway Drive, Suite 300, Hanover, MD 21076 USA.

Objective w_1 calculates the classical UFLP cost of opening facilities and serving customers from their primary facilities. Objective w_2 computes the expected transportation cost, accounting for both normal and disrupted modes. We can then replace the RFLP objective function (9.61) with

$$\text{minimize} \quad \alpha w_1 + (1-\alpha) w_2, \tag{9.77}$$

where α is a parameter specified by the user, $0 \le \alpha \le 1$. Large values of α place more emphasis on objective 1, small values on objective 2. (Setting $\alpha = 1$ is equivalent to solving the UFLP.) The decision maker might select a single value of α, but more commonly, the goal is to generate the *trade-off curve* that depicts the relationship between the two objectives. In essence, the trade-off curve (also known as the *Pareto curve* or *efficient frontier*) tells us how much of one objective we must sacrifice in order to improve the other objective.

How can we generate such a trade-off curve? The brute-force approach would be to simply solve the RFLP (with objective (9.77)) for every value of α between 0 and 1 in increments of, say, 0.001. But it is preferable to use a much more elegant and efficient approach called the *weighting method*. We won't describe the details of this approach; see instead Cohon (1978).

The trade-off curve for the 49-node problem discussed in Section 9.6.1 is pictured in Figure 9.11. Each point represents a different solution to the RFLP, and the axes represent the two objectives. The left-most point is the UFLP solution ($\alpha = 1$). The left portion of the trade-off curve is "steep," indicating that large improvements in reliability can be attained with only small increases in the classical UFLP cost. For example, the third point on the trade-off curve has a 3.1% increase in UFLP cost from the original UFLP solution but a 13.4% decrease in expected disruption cost, and the fourth point has a 7.3% increase in cost but a 26.5% decrease in expected disruption cost.

CASE STUDY 9.1 Disruption Management at Ford

Ford Motor Company operates an enormous supply chain. The company procures materials from over 4000 supplier sites, which in turn procure from hundreds of thousands of suppliers. Ford owns more than 50 facilities, manages 130,000 unique parts, and spends tens of billions of dollars every year procuring these parts. A disruption in such a supply chain can have a major impact on the firm's performance. For example, in 2011, Toyota suffered a loss of output of roughly 150,000 vehicles due to an earthquake in Japan, and 240,000 more due to flooding in Thailand (Toyota Motor Corporation 2012).

Recognizing the importance of supply chain risk management, Ford and researchers from the Massachusetts Institute of Technology (MIT) conducted a 3-year research study to develop and implement models to identify vulnerabilities in Ford's supply chain. Their work is described by Simchi-Levi et al. (2015); we summarize it here.

The approach centers around two models. Both models treat the supply chain as a graph, in which nodes represent parts or processes and edges represent flows among them (much like the multiechelon inventory models of Chapter 6). They are post-event models, meaning that they assume that a disruption has already taken place and that its characteristics are known. However, they can also be used for pre-event planning, as we discuss below. (The disruption models in this chapter are pre-event models; they plan for uncertain disruptions that have not yet occurred.)

The first model takes as input the *time to recover* (TTR), i.e., the time it takes each disrupted element of the graph to fully recover from the disruption. It optimizes the firm's reaction to the disruption by allocating existing inventory and setting production levels. Since the TTR can be difficult to estimate, the second model calculates the *time to survive* (TTS), i.e., the longest disruption of a given node (or nodes) that the supply chain can endure without any loss of performance. The two models are referred to as the TTR and TTS models, respectively.

We will only discuss the TTR model here; see Problem 9.24 for the TTS model. Moreover, we will discuss only a single-echelon version of the TTR model, which considers multiple supply nodes but not the interconnections among them. Simchi-Levi et al. (2015) also formulate a multiechelon model, but we will omit that model here.

Let J be the set of supply nodes, and let K be the set of vehicles (products). The model assumes that we are in a certain disruption scenario, denoted s. Let $J_{ks} \subseteq J$ be the set of nodes that can produce vehicle k in scenario s. Let t_s be the TTR for the disruption in scenario s. Vehicle $k \in K$ has a demand of d_k per unit time, and each vehicle produced earns the firm a profit of π_k. There are s_k units of finished-goods inventory of vehicle k available. Node $j \in N$ has a production capacity of c_j vehicles per unit time. The TTR model has two decision variables: y_{jks} is the number of units of vehicle $k \in K$ produced at node $j \in J$ during the disruption, and l_{ks} is the number of units of lost demand for vehicle k, both indexed by scenario s. The model assumes that lead times are negligible compared to the length of the disruption. It ignores transportation costs, too, focusing instead on production. The TTR model can then be formulated as follows:

$$\text{minimize} \quad \sum_{k \in K} \pi_k l_{ks} \tag{9.78}$$

$$\text{subject to} \qquad l_{ks} \geq d_k t_s - \sum_{j \in J_{ks}} y_{jks} - s_k \qquad \forall k \in K \tag{9.79}$$

$$\sum_{k: j \in J_{ks}} y_{jks} \leq c_j t_s \qquad \forall j \in J \tag{9.80}$$

$$y_{jks}, \; l_{js} \geq 0 \qquad \forall j \in J, \forall k \in K \tag{9.81}$$

The objective function (9.78) calculates the total profit lost due to lost vehicle sales. Constraints (9.79) sets the number of units of lost demand for vehicle k equal to the demand for the vehicle during the TTR minus its supply, i.e., minus the number of units produced and the number of units available in inventory. (The constraint is written as an inequality but will hold as an equality if the lost sales are positive.) Constraints (9.80) enforce the capacity at node j during the disruption. Constraints (9.81) are nonnegativity constraints.

The TTR and TTS models are post-event models: They assume we know what the disruption has affected and how long it will last. Ford develops a set of scenarios s that they are interested in—for example, all scenarios in which a single node is disrupted—and runs the model for each s. This allows Ford to use the models for both strategic and tactical (pre-event) planning. At the strategic level, Ford uses the models to identify parts or suppliers that introduce the most risk exposure into the supply chain, i.e., that have large objective values in the optimization model (9.78)–(9.81). In many cases, the parts with the largest risk exposure turned out to be low-cost, commodity-type parts, rather than high-value components such as engines or instrument panels. (This is not to say that such components are not important, but only that the current supply chain could withstand a disruption to these parts through existing inventory and production capacity.) They can then devote resources to prevent or mitigate these disruptions. Tactically, they use the models to track changes in the risk exposure over time, and to alert planners to investigate further and take corrective action. And, of course, the tool can also be used for operational (post-event) planning by optimizing the firm's reaction after a disruption. For example, the company recently used the tool to reallocate supply in response to political instability in one geographical region.

Through this process, Ford identified 2600 supplier sites that, if disrupted, could cause losses as large as \$2.5 billion, and it prioritizes these sites in its risk-planning process. Moreover, it also identified 400 sites that were previously receiving too *much* of Ford's risk-management resources, allowing it to reallocate those resources to sites with higher risk exposure. One Ford manager described the models as "key game changers," allowing them to allocate their risk-management resources effectively and accurately.

PROBLEMS

9.1 (Disruption-Prone Bicycle Parts) A bicycle manufacturer buys a particular cable used in its bicycles from a single supplier located in South America. The manufacturer follows a periodic-review base-stock policy for its inventory of cables, placing an order with the supplier every week. The supplier occasionally experiences disruptions due to hurricanes, labor actions, and other events. These disruptions follow a Markov process with disruption probability $\alpha = 0.1$ and recovery probability $\beta = 0.4$. When not disrupted, the supplier's lead time is negligible. Cables are used by the manufacturer at a constant rate of 6000 per week. Inventory incurs a holding cost of \$0.002 per cable per week. If the manufacturer runs out of cables, it must delay production, resulting in a cost that amounts to \$0.05 per cable per week.

a) On average, how many weeks per year is the supplier disrupted? On average, how long does each disruption last?

b) What is the optimal base-stock level for cables?

9.2 (Disruption-Prone Appliance Parts) An appliance manufacturer buys a particular gasket used in its dishwashers from a single supplier located in Turkey. The manufacturer follows a periodic-review base-stock policy for its inventory of gaskets, placing an order with the supplier every week. The supplier occasionally experiences disruptions that follow a Markov process with disruption probability $\alpha = 0.05$ and recovery probability $\beta = 0.2$. When not disrupted, the supplier's lead time is negligible. Gaskets are used by the manufacturer at a constant rate of 2400 per week. Inventory incurs a holding cost of \$0.55 per gasket per week. If the manufacturer runs out of gaskets, it must delay production, resulting in a cost that amounts to \$2.75 per gasket per week.

a) On average, how many weeks per year is the supplier disrupted? On average, how long does each disruption last?

b) What is the optimal base-stock level for gaskets?

9.3 (Stocking Latex Gloves) A university health clinic uses exactly 4 boxes of latex gloves per week and orders gloves once per week. It costs the clinic \$0.10 per week to hold one box of gloves in inventory. If the clinic runs out of gloves before the end of the week, they must buy more gloves from a local pharmacy at a cost that is \$0.50 greater per box than the cost their normal supplier charges.

The clinic's normal supplier of gloves is quite unreliable. In any period, there is a 50% chance that the supplier is disrupted and cannot ship any gloves. This probability is independent of whether the supplier was disrupted in the previous period; that is, $\alpha = \beta = 0.5$. What is the optimal base-stock level for the clinic to use when ordering gloves?

9.4 (Random Yield for Steel) Return to Problem 3.1. Suppose that the amount of steel delivered by the supplier differs randomly from the order quantity, and the auto manufacturer must accept whatever quantity the supplier delivers. Let Q be the order quantity.

a) Suppose the delivery quantity is given by $Q + Y$, where $-Y \sim \exp(0.02)$. What is Q^*?

b) Suppose the delivery quantity is given by QZ, where $Z \sim U[0.8, 1.0]$. What is Q^*?

9.5 (Staffing Truck Drivers) The US trucking industry suffers from notoriously high employee turnover, with turnover rates often well in excess of 100% (Paz-Frankel 2006). This makes advance planning difficult since it is difficult to predict how many drivers will be available when needed. Suppose a trucking company needs 25 drivers every day. If the company asks S drivers to report to work on a given day, the number of drivers who actually show up is given by $S + Y$, where $Y \sim U[-5, 0]$. Drivers who report to work but are not needed must still be paid their daily wage of \$150. For each driver fewer than 25 that show up, the company will be unable to deliver a load, incurring a cost of \$1200.

Find S^*, the optimal number of drivers to ask to report to work. (Fractional solutions are acceptable.) Also report $g(S^*)$, the optimal expected cost per day.

9.6 (Simulating Truck-Driver Staffing) Build a simulation model of the truck-driver-staffing problem in Problem 9.5. Simulate the system for at least 1000 periods and assume the company uses $S = 27$ (which is not necessarily optimal). Report the expected cost per day. You may use Excel, MATLAB, or another package or language of your choice.

9.7 (EOQD Approximation) Suppose that, in the EOQD model of Section 9.2.1, we replace ψ (a function of Q) with

$$\hat{\psi} = \frac{\lambda}{\lambda + \mu}$$

(which is independent of Q). Let $\hat{g}$ be the cost function that results from replacing ψ with $\hat{\psi}$ in (9.5). It is known that $\hat{g}$ is convex (you do not need to prove this).

a) Prove that the derivative of $\hat{g}(Q)$ is

$$\hat{g}'(Q) = \frac{\frac{h\mu^2}{2}Q^2 + \hat{\psi}dh\mu Q - (Kd\mu + d^2p\hat{\psi})\mu}{(Q\mu + \hat{\psi}d)^2}.$$

b) Prove that $\hat{Q}^*$, the Q that minimizes $\hat{g}$, is given by

$$\hat{Q}^* = \frac{-\hat{\psi}dh + \sqrt{(\hat{\psi}dh)^2 + 2hd\mu(K\mu + dp\hat{\psi})}}{h\mu}. \tag{9.82}$$

9.8 (Implementing EOQD Approximation) Consider an instance of the EOQD with $K = 35$, $h = 4$, $p = 22$, $d = 30$, $\lambda = 1$, and $\mu = 12$.

a) Find Q^* for this instance using optimization software of your choice. Report the expected cost, $g(Q^*)$.

b) Consider the following heuristic for the EOQD:

1. Set Q equal to the EOQ.
2. Calculate ψ using the current value of Q.
3. Find Q using (9.82) from Problem 9.7, setting $\hat{\psi}$ equal to the current ψ from step 2.
4. If Q has changed more than ϵ since the previous iteration (for fixed $\epsilon > 0$), then go to 2; otherwise, stop.

Using this heuristic and any software package you like, find a near-optimal Q using $\epsilon = 10^{-3}$. Report the Q you found, its cost $g(Q)$, and the percentage difference between $g(Q)$ and $g(Q^*)$ from part (a).

9.9 (Optimal Cost for Base-Stock Policy with Disruptions) In the base-stock problem with disruptions discussed in Section 9.2.2, let R be the smallest n such that $F(n) \geq p/(p + h)$, where $F(x)$ is as defined in (9.10). Prove that

$$g(S^*) \geq d\left[p\sum_{n=R+1}^{\infty} \pi_n n - h\sum_{n=0}^{R} \pi_n n\right],$$

and that equality holds if and only if $F(R) = p/(p + h)$.

9.10 **(Proof of Lemma 9.2)** Prove Lemma 9.2.

9.11 **(Disruptions = Stochastic Demand?)**

a) Develop a *stochastic demand process* that is equivalent to the *stochastic supply process* in the base-stock model with disruptions from Section 9.2.2. In particular, formulate a demand distribution such that, if the demand is iid stochastic following your distribution but the supply is deterministic, the expected cost is equal to the expected cost given by (9.14), assuming we order up to the same S in every period. Prove that the two expected costs are equal. Make sure you specify both the possible values of the demand and the probability of each value, i.e., the pmf.

b) In part (a) you proved that, under the optimal solution, the expected cost is the same in both models. Is the entire distribution of the random variable representing the cost also the same in both models?

9.12 **(Newsvendor with Random Yield and Demand)** The additive yield model in Section 9.3.2.2 assumes the demand is deterministic and equal to d. Suppose instead that the demand is given by a random variable $D \sim N(d, \sigma^2)$. The yield, Y, continues to be random, with a normal distribution, and D and Y are independent. Show that we can solve this problem *either* by defining a new random variable that represents the "net demand" and using the classical newsvendor model, *or* by defining a new random variable that represents the "net yield" and using the model from Section 9.3.2.2. Show that these two approaches are equivalent.

9.13 **(Optimal Cost for EOQ with Additive Yield Uncertainty)** Prove that, in the EOQ model with additive yield uncertainty (Section 9.3.1.2), the optimal expected cost is given by

$$g(Q^*) = h(Q^* + \mathbb{E}[Y]).$$

9.14 **(Optimal Cost for EOQ with Multiplicative Yield Uncertainty)** Prove that, in the EOQ model with multiplicative yield uncertainty (Section 9.3.1.3), the optimal expected cost is given by

$$g(Q^*) = hQ^* \frac{\mathrm{Var}[Z] + \mathbb{E}[Z]^2}{\mathbb{E}[Z]}.$$

9.15 **(EOQ with Discrete Yield Uncertainty)** Suppose that, in the EOQ models with additive and multiplicative yield uncertainty, Y and Z are discrete random variables rather than continuous ones, but that Q may still take continuous values. Show that the expected cost functions (9.22) and (9.24) remain the same, as do the optimal solutions (9.23) and (9.25).

9.16 **(Production Functions)** Provide appropriate definitions of W_i (that is, of B_i and X_i) to model each of the forms of supplier unreliability listed below for the multisupplier newsvendor model of Section 9.4. In each case, assume that Y is a nonnegative random variable with known distribution.

a) The supplier delivers either 90% of the order quantity, or Y, whichever is smaller.

b) If the order is for fewer than Y units, the supplier will deliver the entire order quantity. If the order is for Y units or more, the supplier will deliver 50% of the order quantity.

c) With probability π, the supplier is down and cannot provide any product. With probability $1-\pi$, the supplier is up. When the supplier is up, it delivers $100Y\%$ of the order quantity, where $0 \le \pi \le 1$ and $0 \le Y \le 1$.

9.17 (Multisupplier Model Example) Consider a four-supplier instance of the multi-supplier newsvendor model of Section 9.4. Let $r = 100$, $v = 20$, $p = 200$, $c_1 = 40$, $c_2 = 50$, $c_3 = 60$, and $c_4 = 70$. Let the demand D be distributed as $N(250, 50^2)$. For each part below, choose supplier production functions W_i (i.e., choose B_i and X_i), $i = 1, \dots, 4$, such that the optimal Q_i^* have the desired property, and argue clearly why your production functions produce Q_i^* with that property.

a) $Q_1^*, Q_2^* > 0$ and $Q_3^*, Q_4^* = 0$. (This illustrates Theorem 9.8(a).)

b) One of the suppliers is perfectly reliable and, in the optimal solution, both the perfectly reliable supplier *and* another supplier are active. (This illustrates that, although Theorem 9.8(b) says that all suppliers that are more expensive than the perfectly reliable supplier must be inactive, cheaper suppliers may be active.)

c) For some i and j with $i < j$, $\alpha_j < \mathbb{E}_{X_i}[W_i'(Q_i^*, X_i)]\alpha_i$ and thus, by Theorem 9.8(c), $Q_j^* = 0$.

9.18 (Expected Profit for Multisupplier Model) Prove that the newsvendor's expected profit is given by (9.37).

9.19 (Proof of Special Case of Lemma 9.6(b)) Prove Lemma 9.6(b) for the case in which supplier i has an exogenous production function; that is, $B_i(Q_i, X_i) = X_i$.

9.20 (Optimality Condition for Exogenous Supply) In the multisupplier model of Section 9.4, prove that, if supplier i has an exogenous production function and $Q_i^* > 0$, then

$$\mathbb{E}_{\mathbf{X}}[F(Q_i + W_{T-i})] = \alpha_i, \tag{9.83}$$

where $W_{T-i} = \sum_{j \neq i} W_j(Q_j, X_j)$.

9.21 (Service Level vs. Cost and Reliability) Suppose there is only a single supplier, and assume it has an exogenous production function as defined in Section 9.4. Suppose further that the supplier's production function is characterized by a parameter ρ, with $\partial G(x;\rho)/\partial\rho < 0$, so that as ρ increases, so does the supplier's reliability. (We'll omit the subscript 1 since there is only a single supplier.)

a) Prove that $SL(Q^*)$ increases with α.

b) Prove that $SL(Q^*)$ increases with ρ.

Hint: You may use (9.83).

9.22 (Proof of Theorem 9.10) Prove Theorem 9.10.

9.23 (Facility-Dependent Disruption Probabilities) Suppose we want each facility to have a different disruption probability q_j in the RFLP model from Section 9.6. If we were to use similar logic as the RFLP, the objective function would become very messy since the q^r terms would be replaced by a product of y_{ijr} variables. Develop an alternate formulation for this problem in which the q_j may be different for each j.

a) Write out your formulation. Your formulation must be linear. Define any new notation that you introduce, and explain the objective function and each of the constraints in words.

b) In a short paragraph, discuss the advantages and disadvantages of your formulation versus the original model.

9.24 (Time-to-Survive Model) Using the notation from Case Study 9.1, formulate the time-to-survive (TTS) model, which calculates the maximum amount of time the firm can last without losing demand when scenario s occurs. If you introduce any new notation, define it clearly. Explain the objective function and constraints in words.

CHAPTER 10

THE TRAVELING SALESMAN PROBLEM

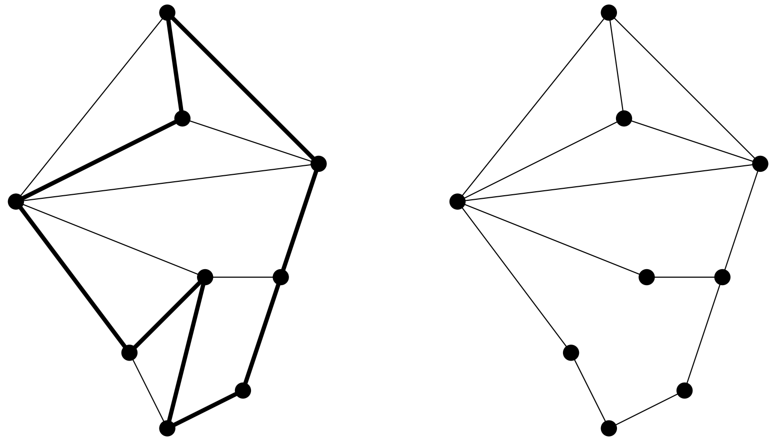
10.1 SUPPLY CHAIN TRANSPORTATION

Transportation is arguably the largest component of total supply chain costs. In 2017, United States businesses spent $966 billion moving freight along roads, rails, waterways, air routes, and pipelines (Council of Supply Chain Management Professionals 2018a). Even small improvements in transportation efficiency can have a huge financial impact. A transportation system has many aspects to optimize, from mode selection to driver staffing to contract negotiations, and mathematical models have been used for all of these issues and more. In this chapter and the next, we cover one important aspect of the transportation-related decisions a firm must make, namely, routing vehicles among the locations they must visit.

We discuss the famous *traveling salesman problem* (TSP) in this chapter. The TSP is important not only because of its practical utility but also because, over the past several decades, the study of the TSP has spurred many fundamental developments in the theory of optimization itself. Then, in Chapter 11, we discuss the *vehicle routing problem* (VRP), which generalizes the TSP by adding constraints that necessitate the use of multiple routes.

Fundamentals of Supply Chain Theory, Second Edition. Lawrence V. Snyder and Zuo-Jun Max Shen.

Companion website: www.wiley.com/go/Snyder/SupplyChainTheory



(a) A Hamiltonian cycle. (b) No Hamiltonian cycle possible.

Figure 10.1 Hamiltonian cycles.

10.2 INTRODUCTION TO THE TSP

10.2.1 Overview

The TSP involves finding the shortest route through n nodes that begins and ends at the same city and visits every node. The TSP is perhaps the best-known combinatorial optimization problem and has been intensely studied by researchers in supply chain management, operations research, computer science, and other fields. Moreover, it serves as the foundation for a great many routing problems, and instances of these problems are solved thousands, if not millions, of times per day by companies and public agencies to plan package and mail deliveries, optimize robot movements, direct naval vessels, fabricate semiconductor chips, and more. (See Applegate et al. (2007) for a thorough discussion of applications of the TSP.)

Closely related to the TSP is the *Hamiltonian cycle problem*, which asks whether there exists a tour on a given graph that visits every node exactly once. For example, a Hamiltonian cycle is marked on the graph in Figure 10.1(a), while the graph in Figure 10.1(b) has *no* Hamiltonian cycle (why?). Finding a Hamiltonian cycle on a general graph is NP-complete (Garey and Johnson 1979). The TSP is the optimization counterpart of the Hamiltonian cycle problem, namely, to find the minimum-cost Hamiltonian cycle; the TSP is therefore NP-hard.

One of the first specific instances of the TSP, and one of the most public, appeared in a 1962 advertisement by Procter & Gamble. The ad offered a $10,000 prize—over $75,000 in today's dollars—for the person who could find the shortest route through the 33 cities in Figure 10.2 for police officers Toody and Muldoon from the popular TV series *Car 54*.[1] The optimal solution, depicted in Figure 10.3, was found by several contestants, though at

[1]A picture of the ad is at `http://www.math.uwaterloo.ca/tsp/gallery/igraphics/car54.html`. The distance matrix for this instance is reported by Karg and Thompson (1964).

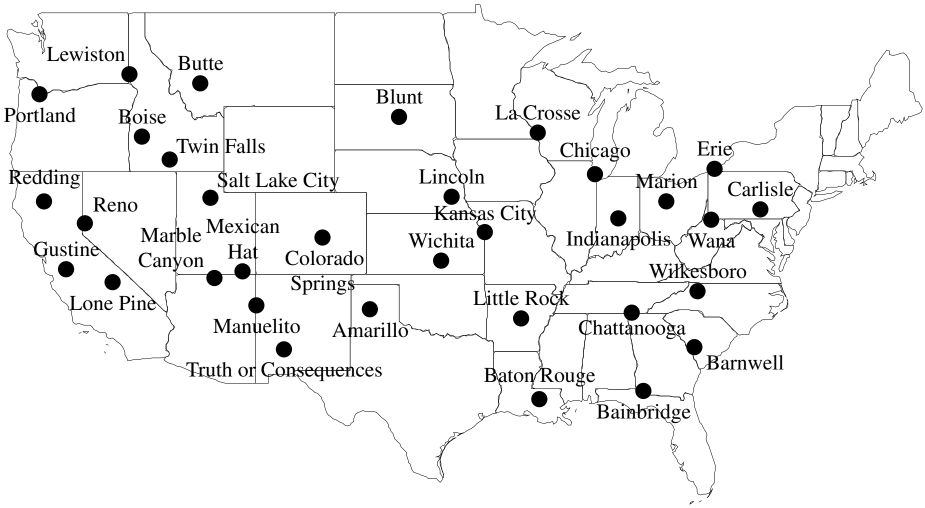


Figure 10.2 *Car 54* TSP instance.

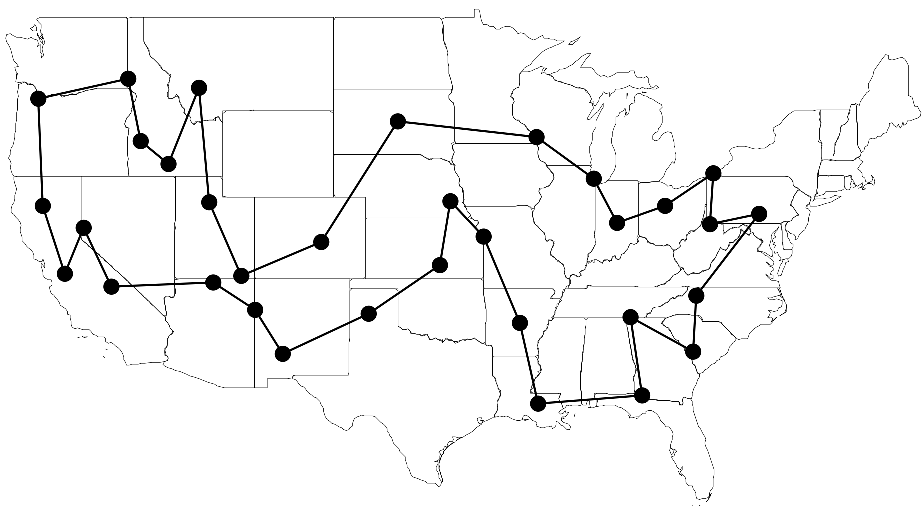


Figure 10.3 Optimal solution to *Car 54* TSP instance. Total distance = 10,861 miles.

the time, it was unknown whether this solution was indeed optimal (Cook 2012). This tour has a total distance of 10,861 miles.

The question "what is the shortest route?" is documented in writings by salespeople as early as 1832, but the problem was not stated formally, or given its name, until the 1940s (Cook 2012). Since then, researchers have developed a wide range of algorithmic methods to solve the TSP—indeed, nearly every exact and heuristic algorithm for discrete optimization has been applied to the TSP at some point—and a large variety of extensions and variants. For a more thorough coverage of this research, see the reviews by Laporte (1992a, 2010), and Hoffman et al. (2013), among others, and the books by Lawler et al. (1985) and Applegate et al. (2007). We also highly recommend the general-interest book by Cook (2012) and its companion website (Cook 2018a), which discuss the TSP's history, algorithms, and allure. Bill Cook's Concorde TSP app for mobile devices implements many exact and heuristic algorithms for the TSP, plus TSP art and games (Cook 2018b). There is even a TSP movie, called *Traveling Salesman* (2012), a political and intellectual thriller in which a group of mathematicians "solve" the TSP, resolve the P vs. NP question, and wrestle with the ethical implications of making their algorithm public.

10.2.2 Formulation of the TSP

Let $N = \{1, \ldots, n\}$ be a set of nodes, and let c_{ij} be the distance (or transportation cost, travel time, etc.) from node i to node j. Many of the properties discussed below rely on the distances satisfying the *triangle inequality*, i.e.,

$$c_{ij} \le c_{ik} + c_{kj} \qquad \forall i, j, k \in N, \tag{10.1}$$

so we will assume this inequality holds. When the triangle inequality holds, the TSP is sometimes referred to as the *metric TSP*. We'll also assume that the distances are symmetric, i.e., that $c_{ij} = c_{ji}$ for all i and j; this means we are considering the *symmetric TSP*.

Let $z(E)$ be the total length of a set E of edges,

$$z(E) = \sum_{(i,j)\in E} c_{ij},$$

and let $z^* = z(\Gamma^*)$ be the length of the optimal TSP tour, Γ^*.

Let x_{ij} be a decision variable that equals 1 if the tour goes from node i to node j *or* from j to i, 0 otherwise. The decision variable x_{ij} is only defined for $i < j$. Since the distances are symmetric, it doesn't matter in which direction the tour is oriented. Here is a *partial* formulation of the TSP:

$$\text{minimize} \quad \sum_{i,j\in N} c_{ij}x_{ij} \tag{10.2}$$

$$\text{subject to} \quad \sum_{i\in N} x_{ih} + \sum_{j\in N} x_{hj} = 2 \qquad \forall h \in N \tag{10.3}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i \in N, \forall j \in N \tag{10.4}$$

The objective function (10.2) calculates the total length of the tour. Constraints (10.3) require the tour to contain exactly two edges that are incident (connected) to node h—in other words, they require the *degree* of node h to equal 2, and they are therefore called *degree constraints*. Constraints (10.4) are integrality constraints. (Technically, we should specify that $i < j$ in the summation in (10.2) and the "for all" part of (10.4), and similarly in the summations in (10.3), but we'll usually omit that extra bit of notation.)

Unfortunately, constraints (10.3)–(10.4) alone are not sufficient to ensure that the resulting solution is a valid tour. To see why, consider the solution depicted in Figure 10.4. Every node has degree 2 in this solution, but it is clearly not a TSP tour because it contains multiple "pieces." These pieces are called *subtours*, and they are at the heart of what makes the TSP difficult.

Our formulation needs to include constraints that prevent subtours—called *subtour-elimination constraints*—and ensure that the solution consists of one contiguous tour. How

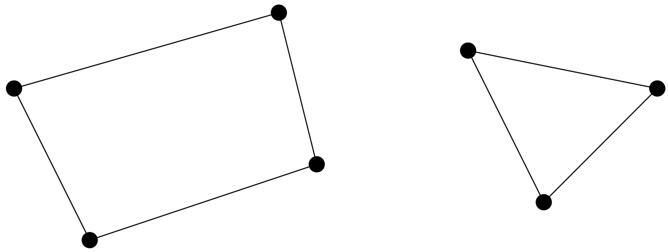


Figure 10.4 Subtours.

can we formulate such constraints? Suppose we have a subset $S \subseteq N$ with at least two nodes. One way to ensure that the edges selected do not contain a subtour on S is to require that there be at least two edges connecting nodes in S with nodes outside of S:

$$\sum_{\substack{i\in S, j\in \bar{S} \text{ or} \\ i\in \bar{S}, j\in S}} x_{ij} \geq 2 \qquad \forall S \subseteq N : 2 \leq |S| \leq n-1, \tag{10.5}$$

where $\bar{S} \equiv N \setminus S$. (Note that there is no constraint for $S = N$, since we *do* want a closed tour on N.) Equivalently, we can limit the number of edges within S to $|S| - 1$:

$$\sum_{i,j\in S} x_{ij} \leq |S| - 1 \qquad \forall S \subseteq N : 2 \leq |S| \leq n-1. \tag{10.6}$$

Both (10.5) and (10.6) are valid subtour-elimination constraints. We'll use (10.6). Both types, unfortunately, consist of $O(2^n)$ individual constraints—we'll discuss a way to deal with this in Section 10.3.3.

Combining (10.2)–(10.4) and (10.6), we get the following formulation for the TSP:

$$\begin{aligned}
\text{(TSP)} \quad \text{minimize} \quad & \sum_{i,j\in N} c_{ij}x_{ij} && && (10.7)\\
\text{subject to} \quad & \sum_{i\in N} x_{ih} + \sum_{j\in N} x_{hj} = 2 && \forall h \in N && (10.8)\\
& \sum_{i,j\in S} x_{ij} \leq |S| - 1 && \forall S \subseteq N : 2 \leq |S| \leq n-1 && (10.9)\\
& x_{ij} \in \{0,1\} && \forall i \in N, \forall j \in N && (10.10)
\end{aligned}$$

We discuss exact algorithms for the TSP in Section 10.3. In Sections 10.4 and 10.5, we discuss construction and improvement heuristics, respectively. In addition to these heuristics, a large variety of metaheuristics have been proposed for the TSP. These include genetic algorithms, tabu search, simulated annealing, and others. Some of these methods produce good solutions, but they typically require longer run times than the construction and improvement heuristics discussed here. For an overview and experimental comparison, see Johnson and McGeoch (1997).

10.3 EXACT ALGORITHMS FOR THE TSP

10.3.1 Dynamic Programming

One of the earliest exact algorithms for the TSP is a dynamic programming (DP) algorithm. Indeed, this was also one of the earliest applications of DP. The approach was proposed independently by Bellman (1962) and by Held and Karp (1962).

Let $S \subseteq N$ be a subset of N that does not contain node 1, and let $j \in S$. Define $\theta(S, j)$ as the length of the shortest route that begins at node 1, visits all nodes in S, and ends at j. If $|S| = 1$, then there is only one such route—the route $1 \to j$—so $\theta(S, j) = c_{1j}$. Suppose $|S| > 1$ and suppose k comes immediately before j on the optimal route from 1 to j through S. Then if we remove j from this route, the resulting route must be the optimal route from 1 to k through S. Therefore, we can calculate $\theta(S, j)$ recursively:

$$\theta(S, j) = \begin{cases} c_{1j}, & \text{if } |S| = 1 \\ \min_{k \in S, k \neq j} \{\theta(S \setminus \{k\}, k) + c_{kj}\}, & \text{otherwise.} \end{cases} \tag{10.11}$$

This algorithm runs in exponential time—roughly $O(n^2 2^n)$. This is not surprising, since the TSP is NP-hard. $O(n^2 2^n)$ is significantly faster than the $O(n!)$ time required for complete enumeration of the solution space, but it still renders this method impractical for all but small instances.

10.3.2 Branch-and-Bound

The first branching algorithm for the TSP was the branch-and-bound algorithm proposed by Little et al. (1963); in fact, their paper was the first to introduce the term *branch-and-bound*. Their algorithm does not solve a relaxation of the TSP to get lower bounds in the branch-and-bound tree. Instead, it calculates lower bounds by modifying the distance matrix c. We'll summarize their approach, and modify it for the symmetric TSP.

Since x_{ij} is defined only for $i < j$, it makes sense to restrict c_{ij} to contain elements for $i < j$ only, as well. Suppose we subtract a constant $\rho \geq 0$ from each element in a given row of this distance matrix. Then the length of any tour Γ will decrease by exactly ρ (why?). In fact, if we subtract a nonnegative constant ρ_i from row i ($i = 1, \ldots, n$) and a nonnegative constant κ_j from column j ($j = 1, \ldots, n$), the length of any tour will decrease by exactly

$$h \equiv \sum_{i=1}^{n} \rho_i + \sum_{j=1}^{n} \kappa_j.$$

Moreover, the optimal tour will not change, only its length. Let c' be the modified distance matrix, i.e., let $c'_{ij} = c_{ij} - \rho_i - \kappa_j$ for $i < j$.

Suppose Γ is a tour. Let $z(\Gamma)$ be the length of the tour under the original matrix and $z'(\Gamma)$ be the length under the modified distance matrix c'. Then

$$z(\Gamma) = z'(\Gamma) + h.$$

If $c'_{ij} \geq 0$ for all $i < j$, then $z'(\Gamma) \geq 0$, and h will be a lower bound on $z(\Gamma)$ for any Γ, and hence, it will be a lower bound on the optimal tour length (under the original distance matrix).

Theorem 10.1 *Let* $c'_{ij} = c_{ij} - \rho_i - \kappa_j$ *for all* $i < j$, *where* $\rho_i, \kappa_j \geq 0$ *are constants such that* $c'_{ij} \geq 0$ *for all* $i < j$. *Then*

$$h \equiv \sum_{i=1}^{n} \rho_i + \sum_{j=1}^{n} \kappa_j$$

is a lower bound on the optimal tour length.

□ EXAMPLE 10.1

Consider the instance pictured in Figure 10.5. Distances c_{ij} are indicated along the edges. If there is no edge between i and j, then c_{ij} equals the shortest-path distance from i to j. The complete distance matrix is given by

$$c = \begin{bmatrix} - & 5 & 21 & 13 & 6 & 15 & 18 & 20 \\ - & - & 16 & 18 & 7 & 12 & 19 & 17 \\ - & - & - & 33 & 16 & 7 & 17 & 11 \\ - & - & - & - & 17 & 26 & 16 & 29 \\ - & - & - & - & - & 9 & 12 & 14 \\ - & - & - & - & - & - & 10 & 5 \\ - & - & - & - & - & - & - & 13 \\ - & - & - & - & - & - & - & - \end{bmatrix}$$

(omitting entries for which $i \geq j$). Let ρ_i equal the minimum value in row i; i.e., $(\rho_1, \ldots \rho_7) = (5, 7, 7, 16, 9, 5, 13)$. Subtracting these constants from the corresponding rows, we get

$$\begin{bmatrix} - & 0 & 16 & 8 & 1 & 10 & 13 & 15 \\ - & - & 9 & 11 & 0 & 5 & 12 & 10 \\ - & - & - & 26 & 9 & 0 & 10 & 4 \\ - & - & - & - & 1 & 10 & 0 & 13 \\ - & - & - & - & - & 0 & 3 & 5 \\ - & - & - & - & - & - & 5 & 0 \\ - & - & - & - & - & - & - & 0 \\ - & - & - & - & - & - & - & - \end{bmatrix}$$

Now let κ_j equal the minimum value in column j of the new matrix: $(\kappa_1, \ldots, \kappa_7) = (0, 9, 8, 0, 0, 0, 0)$. Subtracting these constants from the corresponding rows, we get

$$c' = \begin{bmatrix} - & 0 & 7 & 0 & 1 & 10 & 13 & 15 \\ - & - & 0 & 3 & 0 & 5 & 12 & 10 \\ - & - & - & 18 & 9 & 0 & 10 & 4 \\ - & - & - & - & 1 & 10 & 0 & 13 \\ - & - & - & - & - & 0 & 3 & 5 \\ - & - & - & - & - & - & 5 & 0 \\ - & - & - & - & - & - & - & 0 \\ - & - & - & - & - & - & - & - \end{bmatrix}$$

Since all elements in c' are nonnegative, we get a lower bound of $\sum_i \rho_i + \sum_j \kappa_j = 79$.

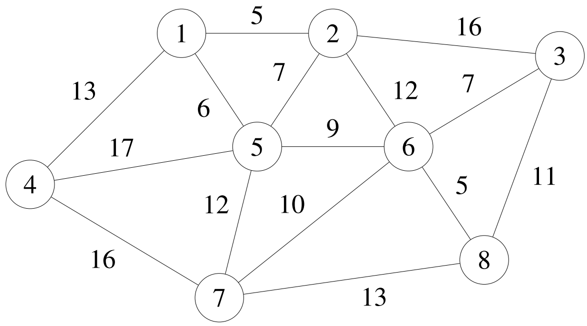


Figure 10.5 TSP instance for examples. If no edge is present between nodes i and j, then c_{ij} equals the shortest-path distance from i to j.

□

The branch-and-bound algorithm of Little et al. (1963) branches on the x_{ij} variables. The choice of which (i, j) to branch on is determined using a simple estimate of the increase in route length that would result from forcing $x_{ij} = 0$. Each time we force an edge into or out of the solution, we can remove a row and column from c'. At a "leaf" of the branch-and-bound tree, c' has only two rows and columns, and the entire corresponding tour can easily be reconstructed. This is the mechanism by which the algorithm obtains upper bounds.

This algorithm is an interesting example of a non–LP-based branch-and-bound method, but it does not play a major role in the way the TSP is solved today.

10.3.3 Branch-and-Cut

A *2-matching* is a set M of edges such that every node in N is contained in exactly two edges. (A 2-matching is a generalization of a *perfect matching*, which is a set of edges such that every node is contained in exactly one edge.) In other words, a 2-matching is a solution to the TSP formulation without the subtour elimination constraints (10.9). The formulation (10.7)–(10.10) without (10.9) is therefore known as the *2-matching relaxation*, and it provides a lower bound on the length of the optimal TSP tour.

Dantzig et al. (1954, 1959) proposed a method for solving the TSP that involved solving the LP relaxation of the 2-matching relaxation, and then manually adding violated subtour-elimination constraints and integrality constraints to coax the solution toward feasibility. Their approach laid the foundation for the *branch-and-cut* method that is ubiquitous in modern methods for solving the TSP, as well as a huge variety of other NP-hard problems. As Cook and Chvátal (2010) put it, "All successful TSP solvers echo their breakthrough. This was the Big Bang."

In a branch-and-cut algorithm, we add *cutting planes* at one or more nodes of the search tree to tighten the LP relaxation. A cutting plane (or simply a *cut*) is a constraint that is satisfied by the optimal integer solution to an optimization problem but not by the optimal solution to its LP relaxation. Adding the cut to the LP relaxation makes the original LP solution infeasible and shrinks the feasible region, thus tightening the LP bound. If we add enough—and good enough—cuts, the LP feasible region will approximate or even equal the convex hull of the IP in the region of the optimal IP solution. In that case, solving the LP relaxation will be equivalent to solving the IP.

For example, consider the following integer programming problem:

$$\begin{array}{lrl} \text{maximize} & -x_1 + 3x_2 & \\ \text{subject to} & x_1 + 2x_2 & \geq 4 \\ & -6x_1 + 10x_2 & \leq 36 \\ & 4x_1 + 2x_2 & \leq 25 \\ & x_1, \quad x_2 & \in \mathbb{Z}_+ \end{array} \tag{10.12}$$

($\mathbb{Z}_+$ is the set of all nonnegative integers.) The optimal solution to (10.12) is $x^* = (3, 5)$, with objective value $z^* = 12$. Its LP relaxation, on the other hand, has optimal solution $x^{LP} = (3.42, 5.65)$, with objective value $z^{LP} = 13.54$. The LP therefore provides a reasonably tight upper bound, but not tight enough to prove the optimality of x^*. The constraints of (10.12) are plotted in Figure 10.6(a), as is the objective function (as a thinner line). The feasible region of the LP relaxation is shaded, and the feasible integer points are the integer points within the shaded region.

Now consider the following constraint:

$$-x_1 + 2x_2 \leq 7. \tag{10.13}$$

Constraint (10.13) is a cut, because it is satisfied by every feasible integer solution for (10.12) (see the dashed line in Figure 10.6(b)) but not by every feasible solution for the LP relaxation of (10.12). In particular, x^{LP} violates (10.13). Therefore, adding (10.13) to the problem shrinks the LP feasible region, and thus tightens the LP bound. The new LP solution is $x^{LP} = (3.6, 5.3)$, with $z^{LP} = 12.3$.

Adding the cut (10.13) does not quite reduce the LP feasible region to the convex hull of the IP feasible set near the optimal solution. But adding one more cut will do the trick:

$$x_1 + x_2 \leq 8. \tag{10.14}$$

Now we have $x^{LP} = (3, 5) = x^*$. (See Figure 10.6(c).) In other words, solving the LP relaxation is equivalent to solving the IP itself.

This raises the question: How do we identify good cuts, i.e., constraints that "separate" the current LP solution? The problem of finding such a cut is called the *separation problem*. In most cutting-plane methods, including those for the TSP, researchers first identify *families* of cuts—that is, they identify structures of the cuts but not their coefficients—and then develop algorithms to solve the separation problem—that is, to determine coefficients so that the cuts will render the current LP solution infeasible. In some cases, the separation problem can be solved efficiently (in polynomial time) and exactly, while in others it must be solved heuristically. The separation problem is typically solved at each node of the search tree to generate cuts that tighten the LP relaxations. Through a combination of branching and adding cuts, branch-and-cut guides the solution to the LP relaxation toward integer feasibility.

Several families of cuts are used in modern branch-and-cut algorithms for the TSP. One family consists of the subtour-elimination constraints. As we noted in Section 10.2.2, there are too many subtour-elimination constraints to include all of them explicitly in the formulation. Therefore, we generate violated subtour-elimination constraints on the fly during the branch-and-bound process. The separation problem—i.e., the problem of

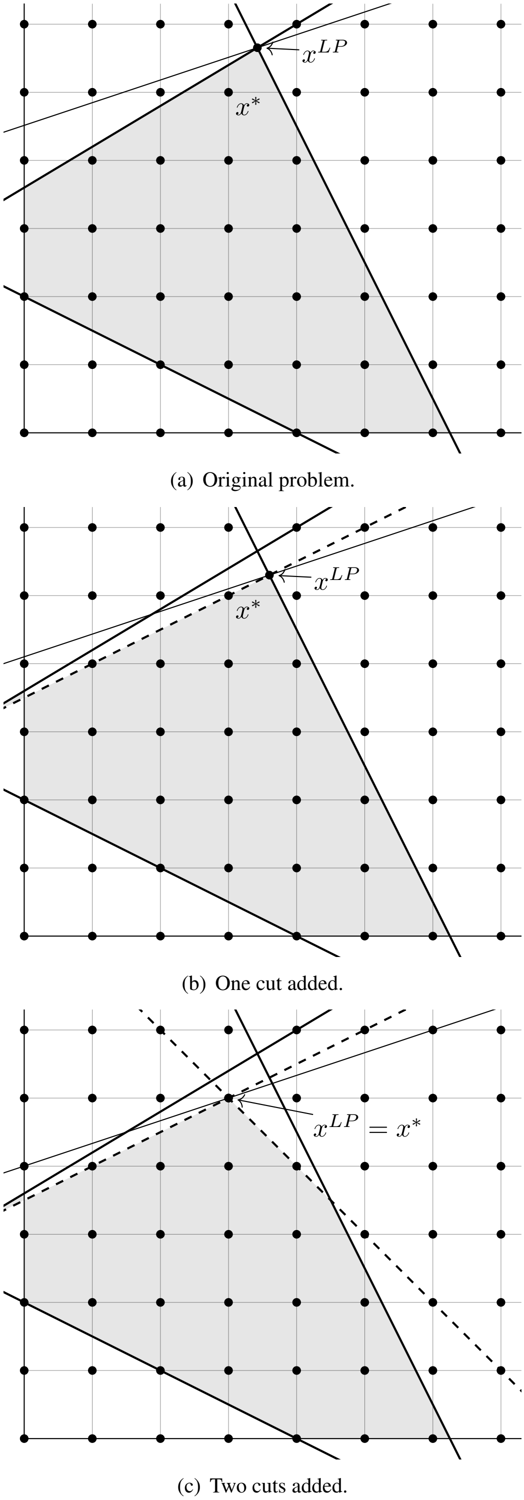


Figure 10.6 Cutting planes.

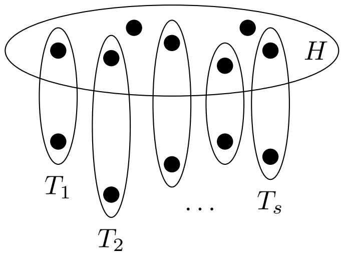


Figure 10.7 Handle and teeth for 2-matching inequality.

deciding *which* constraints to add on the fly—can be solved in polynomial time (Crowder and Padberg 1980, Padberg and Rinaldi 1990a). The basic idea is as follows: Suppose we solve the LP relaxation of (10.7)–(10.10) with (10.9) replaced by the equivalent constraints (10.5), but only some of these constraints included. Let x^{LP} be the optimal solution. We wish to find constraints (10.5) that are violated by x^{LP}. The *support graph* of this solution has node set N and edge set $\{(i,j)|x_{ij}^{LP} > 0\}$, with weight x_{ij}^{LP} on edge (i,j). There is a violated constraint (10.5) if and only if there is an $S \subset N$ such that the total weight on the edges out of S is less than 2—in other words, if there is a cut in the support graph whose weight is less than 2. Therefore, finding a violated subtour-elimination constraint is equivalent to solving a minimum-cut problem, which can be done in polynomial time.

A second family of cuts are called *2-matching inequalities*, or *Blossom inequalities* (Hong 1972). Suppose we have a set $H \subseteq N$ and s pairwise disjoint sets $T_1, \ldots, T_s \subseteq N$ such that each T_j contains exactly two nodes, one in H and one not in H, and such that s is odd and at least 3. (See Figure 10.7 for an illustration.) Anticipating the comb inequalities discussed next, H is called the *handle* and the T_k are called *teeth*. A 2-matching inequality has the form

$$\sum_{i,j\in H} x_{ij} + \sum_{k=1}^{s} \sum_{i,j\in T_k} x_{ij} \le |H| + \frac{1}{2}(s-1). \tag{10.15}$$

In other words, the number of tour edges contained within the handle plus the number of edges contained within teeth cannot exceed $|H| + \frac{1}{2}(s-1)$. The next theorem confirms that (10.15) is a valid inequality for the TSP, i.e., a cut.

Theorem 10.2 *For any handle $H \subseteq N$ and teeth $T_1, \ldots, T_s \subseteq N$ such that*

- *each T_k contains exactly one node in H and one node not in H,*
- *$T_1, \ldots, T_s$ are pairwise disjoint, and*
- *$s \ge 3$ and odd,*

the 2-matching inequality (10.15) *is valid for every tour through N.*

Proof. Suppose, for a contradiction, that we have a tour such that

$$\sum_{i,j\in H} x_{ij} + \sum_{k=1}^{s} \sum_{i,j\in T_k} x_{ij} > |H| + \frac{1}{2}(s-1).$$

It suffices to assume that the violation is as small as possible, i.e., that

$$\sum_{i,j \in H} x_{ij} + \sum_{k=1}^{s} \sum_{i,j \in T_k} x_{ij} = |H| + \frac{1}{2}(s+1).$$

The sum of the degrees of the nodes in H is at least

$$D \equiv 2 \sum_{i,j \in H} x_{ij} + \sum_{k=1}^{s} \sum_{i,j \in T_k} x_{ij}.$$

("At least" because there may be additional edges incident to the nodes in H, in addition to those contained in H or the teeth.) D is minimized when as many of the $|H| + \frac{1}{2}(s+1)$ edges as possible are in the teeth, i.e., when

$$\sum_{k=1}^{s} \sum_{i,j \in T_k} x_{ij} = s$$
$$\sum_{i,j \in H} x_{ij} = |H| - \frac{1}{2}s + \frac{1}{2}.$$

Therefore,

$$D \geq 2\left(|H| - \frac{1}{2}s + \frac{1}{2}\right) + s = 2|H| + 1.$$

Since D is a lower bound on the sum of the degrees of the nodes in H, at least one node in H has degree greater than 2. This contradicts our assumption that x defines a tour. ■

2-matching inequalities can be written in various other forms. For example, the next proposition gives an equivalent inequality.

Proposition 10.3 *For any handle $H \subseteq N$ and teeth $T_1, \ldots, T_s \subseteq N$ satisfying the conditions of Theorem 10.2, the following inequality is valid for every tour through N:*

$$\sum_{\substack{i \in H \\ j \notin H}} x_{ij} + \sum_{k=1}^{s} \sum_{\substack{i \in T_k \\ j \notin T_k}} x_{ij} \geq 3s + 1. \tag{10.16}$$

Proof. Let Ω be the set of teeth such that the tour travels directly from one node in the tooth to the other; i.e., $\Omega = \{T_k = (i,j) | x_{ij} = 1\}$. Then

$$\sum_{\substack{i \in H \\ j \notin H}} x_{ij} \geq |\Omega| \tag{10.17}$$

by the definitions of H and T_k. In fact, if $|\Omega| = s$, then

$$\sum_{\substack{i \in H \\ j \notin H}} x_{ij} \geq s + 1 \tag{10.18}$$

since the left-hand side is even (for every edge that exits H, there must be another that enters it) but s is odd.

Moreover, a given tooth T_k either has two edges coming out of it (if $T_k \in \Omega$) or four edges coming out of it (otherwise). Thus,

$$\sum_{k=1}^{s} \sum_{\substack{i \in T_k \\ j \notin T_k}} x_{ij} = 2|\Omega| + 4(s - |\Omega|) = 4s - 2|\Omega|. \tag{10.19}$$

Therefore, if $|\Omega| = s$,

$$\sum_{\substack{i \in H \\ j \notin H}} x_{ij} + \sum_{k=1}^{s} \sum_{\substack{i \in T_k \\ j \notin T_k}} x_{ij} \geq s + 1 + 2s = 3s + 1$$

by (10.18) and (10.19), while if $|\Omega| \leq s - 1$,

$$\sum_{\substack{i \in H \\ j \notin H}} x_{ij} + \sum_{k=1}^{s} \sum_{\substack{i \in T_k \\ j \notin T_k}} x_{ij} \geq |\Omega| + 4s - 2|\Omega| \geq 4s - (s-1) = 3s + 1$$

by (10.17) and (10.19). ■

The separation problem for 2-matching inequalities can be solved in polynomial time. Padberg and Rao (1982) provided the first such algorithm, which relies on solving max-flow problems on a specialized graph. Subsequent improvements to this algorithm were proposed by Grötschel and Holland (1987) and Letchford et al. (2004).

If the teeth are allowed to have more than two nodes each—but at least one node in H and one node not in H—then we have a *comb*, as depicted in Figure 10.8. Combs give rise to the following *comb inequalities*:

$$\sum_{i,j \in H} x_{ij} + \sum_{k=1}^{s} \sum_{i,j \in T_k} x_{ij} \leq |H| + \sum_{k=1}^{s} (|T_k| - 1) + \frac{1}{2}(s-1). \tag{10.20}$$

As with 2-matching inequalities (10.15), the left-hand side represents the number of edges of the tour that are contained within the handle or within the teeth. Comb inequalities are valid inequalities, as the next theorem attests.

Theorem 10.4 *For any handle $H \subseteq N$ and teeth $T_1, \ldots, T_s \subseteq N$ such that*

- *each T_k contains at least one node in H and one node not in H,*
- *$T_1, \ldots, T_s$ are pairwise disjoint, and*
- *$s \geq 3$ and odd,*

the comb inequality (10.20) *is valid for every tour through N.*

Proof. Omitted; see Problem 10.10. ■

Comb inequalities can also be written in other forms; for example, (10.16) is valid for comb inequalities, as well. (See Problem 10.11.)

Grötschel and Padberg (1979) prove that comb inequalities and subtour-elimination constraints together are *facet-defining*, which means that if we add all possible comb

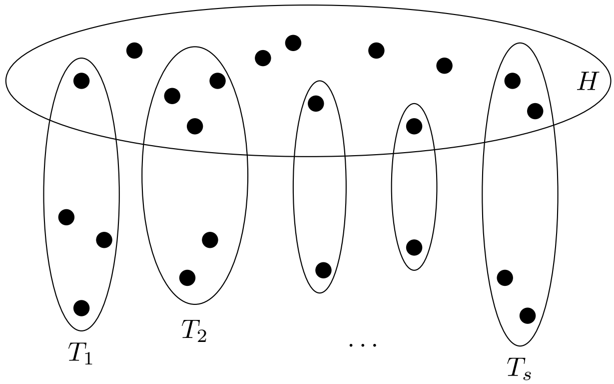


Figure 10.8 Handle and teeth for comb inequality.

inequalities to the feasible region of the LP relaxation of (10.7)–(10.10), the LP relaxation will equal the convex hull of the IP, and solving the LP relaxation will be equivalent to solving the TSP itself. Unfortunately, there is no known polynomial-time exact separation algorithm for comb inequalities, nor is it known whether the separation problem is NP-hard (Applegate et al. 2007). Therefore, the problem is generally solved heuristically; see, e.g., Padberg and Rinaldi (1990b), Grötschel and Holland (1991), and Applegate et al. (1995).

Many other types of cuts have been proposed for the TSP. These include clique-tree (Grötschel and Pulleyblank 1986), path (Cornuéjols et al. 1985), star (Fleischmann 1988), hypohamiltonian (Grötschel 1980a), chain (Padberg and Hong 1980), and ladder inequalities (Boyd et al. 1995). See Applegate et al. (2007) for an in-depth discussion of the branch-and-cut approach for the TSP, including a detailed description of how branch-and-cut is implemented in the Concorde TSP solver (Applegate et al. 2006), widely recognized as the most powerful exact TSP solver available.

10.4 CONSTRUCTION HEURISTICS FOR THE TSP

In this section, we discuss construction heuristics for the TSP. For some heuristics, we will prove that, for all instances,

$$\frac{z^H}{z^*} \le \eta, \tag{10.21}$$

where z^H is the objective value of the solution returned by a given heuristic H, z^* is the optimal objective value, and η is a constant. Equation (10.21) provides a fixed *worst-case error bound.* If a heuristic executes in polynomial time (as most heuristics do, otherwise an exact algorithm may be preferable) and a bound of the form in (10.21) exists, the heuristic is called an *approximation algorithm*, or sometimes an η*-approximation algorithm*.

It is nice when such bounds are available, since then we have a guarantee on the performance of the heuristic. Unfortunately, not all heuristics have fixed worst-case bounds—some heuristics may return solutions that are arbitrarily far from the optimal solution. In fact, the situation is even worse: For general distance matrices (for which the triangle inequality need not hold), there are no polynomial-time approximation algorithms, unless P = NP (Sahni and Gonzalez 1976):

Theorem 10.5 *Suppose there exists a polynomial-time heuristic H and a constant $\eta \geq 1$ such that*

$$\frac{z^H}{z^*} \leq \eta \tag{10.22}$$

for all instances of the traveling salesman problem. Then P = NP.

Proof. Suppose, for a contradiction, that there is a polynomial-time heuristic H and a constant $\eta \geq 1$ for which (10.22) holds for all instances. We will use this heuristic to solve the Hamiltonian cycle problem. Since the Hamiltonian cycle problem is NP-complete and the heuristic runs in polynomial time, we must have P = NP.

Let N and E be the sets of nodes and edges, respectively, in an arbitrary instance of the Hamiltonian cycle problem. Consider an instance of the TSP with nodes N and distances

$$c_{ij} = \begin{cases} 1, & \text{if } (i,j) \in E \\ n\eta, & \text{otherwise.} \end{cases}$$

If a Hamiltonian cycle exists, then $z^* = n$, where z^* is the optimal objective value of the TSP instance. If $z^* = n$, then by (10.22), $z^H \leq n\eta$. On the other hand, if there is no Hamiltonian cycle, then $z^* \geq n\eta + n - 1 > n\eta$, and since $z^H \geq z^*$, we have $z^H > n\eta$. In other words, $z^H \leq n\eta$ if and only if a Hamiltonian cycle exists. Therefore, H is a polynomial-time algorithm to solve the Hamltionian cycle problem. ∎

Note that the distance matrix in the proof of Theorem 10.5 may not satisfy the triangle inequality. This is an important point, because, as we will see, there are polynomial-time heuristics that have fixed worst-case bounds if the triangle inequality holds.

We now discuss several construction heuristics. The heuristics in Sections 10.4.1–10.4.5 are called *insertion heuristics* because they begin with an empty tour and iteratively insert nodes onto the tour. Much the analysis in those sections derives from Rosenkrantz et al. (1977). The heuristics in Sections 10.4.6 and 10.4.7, in contrast, begin by constructing a minimum spanning tree and then building a tour based on the tree.

10.4.1 Nearest Neighbor

We begin with a very simple heuristic called the *nearest neighbor* (NN) heuristic. The heuristic begins with an arbitrary node. At each iteration, we add the unvisited node that is closest to the current node (and set that node as the new current node). When all nodes have been added, we return to the starting node. The NN heuristic executes in $O(n^2)$ time, where n is the number of nodes.

Algorithm 10.1 lists pseudocode for the NN heuristic. In the pseudocode, Θ represents the set of nodes that are on the current tour and $\hat{\imath}$ represents the current node.

☐ **EXAMPLE 10.2**

Consider again the instance in Figure 10.5. Let's begin the NN heuristic at node 1. The nearest neighbor to node 1 is node 2, so we add it, followed by 5. The nearest neighbor to 5 that is not already on the tour is 6. (Nodes 1 and 2 are nearer to 5, but they are already on the tour.) Continuing in this way, we add nodes 8, then 3, then 7, then 4, and finally back to 1. The complete tour is $1 \to 2 \to 5 \to 6 \to 8 \to 3 \to 7 \to 4$, with total distance 83. (See Figure 10.9.)

Algorithm 10.1 Nearest neighbor heuristic

choose arbitrary node i_0; $\Theta \leftarrow \{i_0\}$; $\hat{\imath} \leftarrow i_0$ ▷ Initialization
while $\Theta \neq N$ **do** ▷ Main loop
    $k^* \leftarrow \mathrm{argmin}_{k \in N \setminus \Theta}\{c_{\hat{\imath}k}\}$ ▷ Closest node to current node
    add edge $(\hat{\imath}, k^*)$ to tour ▷ Add k^* to tour
    $\hat{\imath} \leftarrow k^*$; $\Theta \leftarrow \Theta \cup \{k^*\}$
end while
add edge $(\hat{\imath}, i_0)$ to tour ▷ Return to start node
return tour

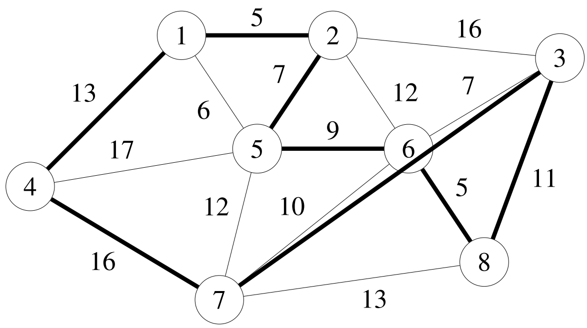


Figure 10.9 Nearest-neighbor tour beginning at node 1 for the example instance in Figure 10.5. Total distance = 83.

□

Clearly, the NN heuristic can easily get "boxed into a corner" from which the only escape is an unattractively long edge (such as $(3, 7)$ in Example 10.2). The NN tour falls into similar traps in the *Car 54* instance, as shown in Figure 10.10. It has a total distance of 13,044 miles (compared to the optimal tour's distance of 10,861). Although Rosenkrantz et al. (1977) report good performance of NN on a small computational experiment, subsequent experiments have reported worse performance compared to other insertion methods (Golden and Stewart 1985).

We now examine the theoretical worst-case behavior of NN.

Theorem 10.6 *Consider an n-node instance of the TSP that satisfies the triangle inequality. Let z^* and z^{NN} be the length of the optimal tour and the nearest-neighbor tour, respectively, for this instance. Then*

$$\frac{z^{NN}}{z^*} \leq \frac{1}{2}\left(\lceil \log_2(n) \rceil + 1\right). \tag{10.23}$$

Moreover, for any $m > 3$, there exists an instance with $n = 2^m - 1$ nodes such that

$$\frac{z^{NN}}{z^*} > \frac{1}{3}\left(\lceil \log_2(n+1) \rceil + \frac{4}{3}\right). \tag{10.24}$$

Proof. Omitted; see Rosenkrantz et al. (1977). ■

The first part of Theorem 10.6 seems like good news, since (10.23) seems like the type of bound we try to prove. However, the bound in the right-hand side of (10.23) is not

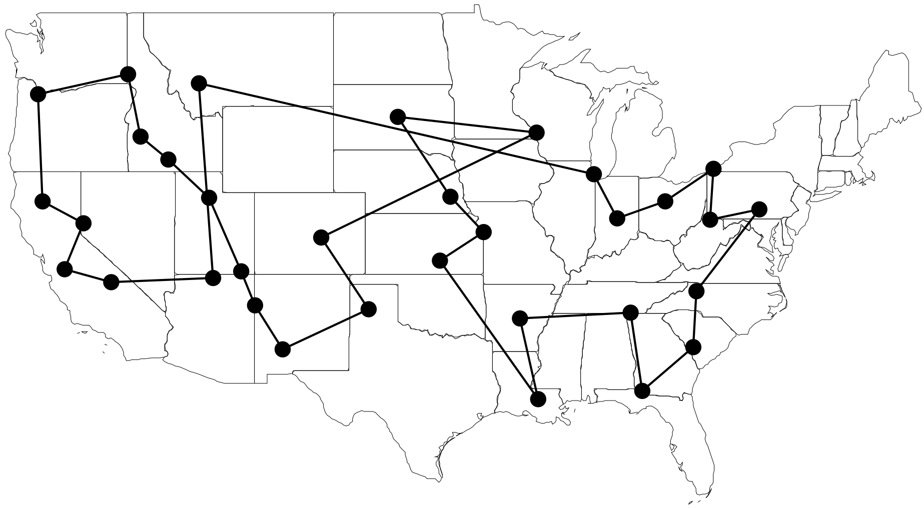


Figure 10.10 Nearest-neighbor solution to *Car 54* TSP instance. Total distance = 13,044 miles.

fixed—it increases with n. Still, this is just an upper bound; maybe the actual ratio doesn't increase with n, leaving open the possibility that some future researcher will prove a fixed worst-case bound. Alas, the second part of the theorem dashes these hopes by proving that we can find instances with arbitrarily bad performance.

10.4.2 Nearest Insertion

The *nearest insertion* (NI) heuristic works as follows: We begin with a tour consisting of an arbitrary node. At each subsequent iteration, we find the unvisited node k^* that is closest to a node m on the tour; node k^* will be the next node to insert onto the tour. If the tour only contains one node thus far, we insert k^* after it; otherwise, we find the tour edge (i^*, j^*) that minimizes the net change in the tour length due to the insertion, defined by

$$\Delta_{ijk} = c_{ik} + c_{kj} - c_{ij}, \tag{10.25}$$

and replace edge (i^*, j^*) with edges (i^*, k^*) and (k^*, j^*). (Note that m, the closest tour node to k^*, will not necessarily equal the i^* or j^* that minimize Δ_{ijk}.) We continue in this way until all cities have been inserted onto the tour. The NI heuristic executes in $O(n^2)$ time. The heuristic is described in pseudocode in Algorithm 10.2.

☐ **EXAMPLE 10.3**

Consider again the example in Figure 10.5. Starting at node 1, the NI heuristic proceeds as follows: The nearest node to 1 is 2, so we have the tour $1 \to 2$. The closest node to $\{1, 2\}$ is node 5, and it doesn't matter whether we insert it before or after node 2 (it just changes the direction of the tour), so we'll insert it before, to get the tour $1 \to 5 \to 2$. The closest nontour node to a tour node is now node 6. The three possible insertion edges give the following net changes:

$$(1, 5): \quad \Delta_{156} = 15 + 9 - 6 = 18$$

Algorithm 10.2 Nearest insertion heuristic

choose arbitrary node i_0; $\Theta \leftarrow \{i_0\}$ ▷ Initialization
while $\Theta \neq N$ **do** ▷ Main loop
  for all $k \in N \setminus \Theta$ **do**
    $c_k^* \leftarrow \min_{m \in \Theta}\{c_{mk}\}$ ▷ Distance from k to tour
  end for
  $k^* \leftarrow \operatorname{argmin}_{k \in N \setminus \Theta}\{c_k^*\}$ ▷ Closest node to tour
  if $|\Theta| = 1$ **then**
    add edges (i_0, k^*) and (k^*, i_0) to tour ▷ Insert k^* as second node
  else
    for all edges (i, j) on tour **do**
      $\Delta_{ijk^*} \leftarrow c_{ik^*} + c_{k^*j} - c_{ij}$ ▷ Net change due to insertion
    end for
    $(i^*, j^*) \leftarrow \operatorname{argmin}_{(i,j) \text{ on tour}}\{\Delta_{ijk^*}\}$ ▷ Best insertion edge
    remove (i^*, j^*) from tour; add (i^*, k^*) and (k^*, j^*) to tour ▷ Insert k^*
  end if
  $\Theta \leftarrow \Theta \cup \{k^*\}$
end while
return tour

$$
\begin{aligned}
(5,2): &\quad \Delta_{526} = 9 + 12 - 7 = 14 \\
(2,1): &\quad \Delta_{216} = 12 + 15 - 5 = 22
\end{aligned}
$$

Therefore, we insert node 6 into the edge $(5, 2)$ to get the tour $1 \to 5 \to 6 \to 2$. Continuing in this way, we make the following insertions:

- Insert 8 into (5,6); $\Delta_{568} = 10$; tour = $1 \to 5 \to 8 \to 6 \to 2$
- Insert 3 into (6,2); $\Delta_{623} = 11$; tour = $1 \to 5 \to 8 \to 6 \to 3 \to 2$
- Insert 7 into (5,8); $\Delta_{587} = 11$; tour = $1 \to 5 \to 7 \to 8 \to 6 \to 3 \to 2$
- Insert 4 into (5,7); $\Delta_{574} = 21$; tour = $1 \to 5 \to 4 \to 7 \to 8 \to 6 \to 3 \to 2$

(See Figure 10.11.) The total distance of this tour is 85.

□

The NI tour in Example 10.3 is actually longer than the NN tour in Example 10.2 (even though it *looks* more reasonable). In general though, NI is a more effective heuristic than NN. For example, the NI tour for the *Car 54* instance, shown in Figure 10.12, has a total distance of 12,588 miles, compared to 13,044 for NN (and 10,861 for the optimal tour).

In fact, unlike NN, NI has a fixed worst-case performance bound:

Theorem 10.7 *For every instance of the TSP that satisfies the triangle inequality,*

$$\frac{z^{NI}}{z^*} \leq 2, \tag{10.26}$$

where z^* and z^{NI} are the lengths of the optimal tour and the nearest-insertion tour, respectively, for this instance.

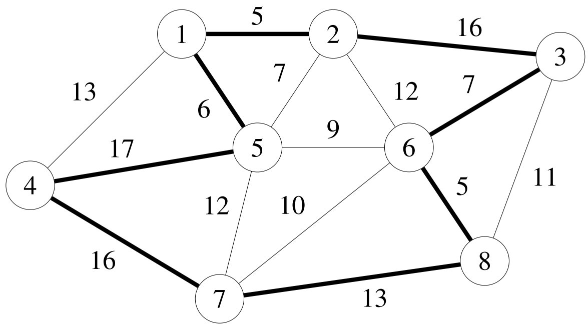


Figure 10.11 Nearest-insertion tour beginning at node 1 for example instance in Figure 10.5. Total distance = 85.

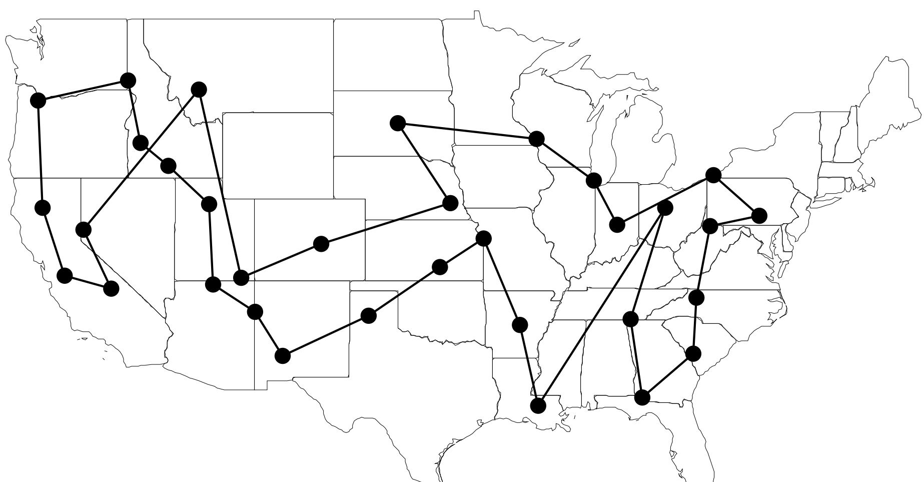
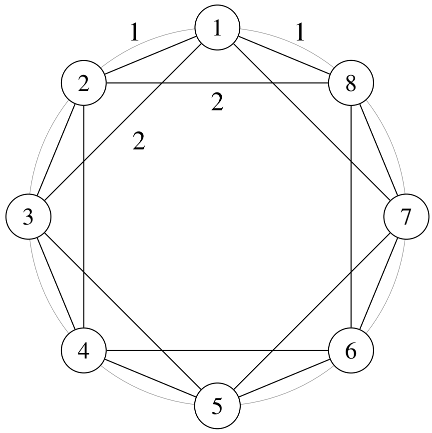
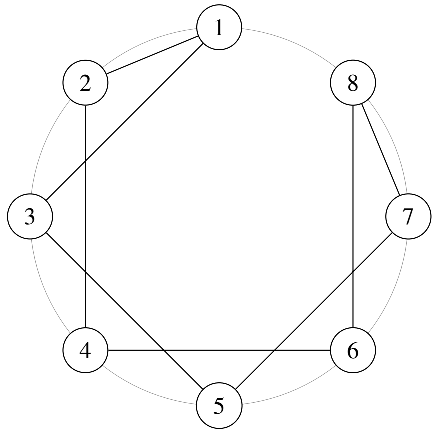


Figure 10.12 Nearest-insertion solution to *Car 54* TSP instance. Total distance = 12,588 miles.



(a) Original network.



(b) Nearest-insertion tour.

Figure 10.13 TSP instance for proof that NI bound is tight.

Proof. Omitted; see Rosenkrantz et al. (1977). ■

In practice, the ratio z^{NI}/z^* is, of course, often much smaller than 2. For example, the bound for the *Car 54* instance is $12,588/10,861 = 1.16$. This raises the question: Is it possible to prove a smaller fixed worst-case bound on z^{NI}/z^* that applies to all instances? The answer is "no"; the bound of 2 is *tight*, as the next proposition demonstrates.

Theorem 10.8 *The bound of 2 in Theorem 10.7 is tight.*

Proof. Consider an n-node instance in which the nodes are spaced evenly along the perimeter of a circle. (An instance with $n = 8$ is shown in Figure 10.13(a).) Edges between consecutive nodes have length 1; edges between nodes that are separated by one node have length 2; and there are no other edges (or, if you prefer, their lengths are ∞). Clearly, the optimal tour simply goes consecutively around the circle, with a total distance of $z^* = n$.

Now consider the nearest-insertion tour. Let's begin with the tour $1 \to 2$. Nodes 3 and n are both closest to nodes on the tour; we'll pick 3 and insert it after node 2. Next, 4 is a nearest node. We cannot insert it into edge $(1, 2)$ because there is no edge $(1, 4)$, nor can we insert it into $(1, 3)$ for the same reason, so we must insert it into $(2, 3)$, to obtain the tour $1 \to 2 \to 4 \to 3 \to 1$. Similarly, we insert node 5 into edge $(3, 4)$, and so on, to obtain the tour in Figure 10.13(b). This tour has $n - 2$ "long" edges and 2 "short" edges, for a total length of $2n - 2$. Therefore,

$$\frac{z^{NI}}{z^*} = \frac{2n-2}{n},$$

which approaches 2 as $n \to \infty$. ■

Note that Theorem 10.8 says that no better bound is possible for the NI heuristic. It does *not* say that no heuristic can possibly obtain a better bound. Indeed, Christofides' heuristic, discussed in Section 10.4.7, has a fixed worst-case bound of 3/2.

A variant of the NI heuristic, called the *cheapest insertion* (CI) heuristic, searches over all k not on the tour and all edges (i, j) on the tour and chooses the insertion with the

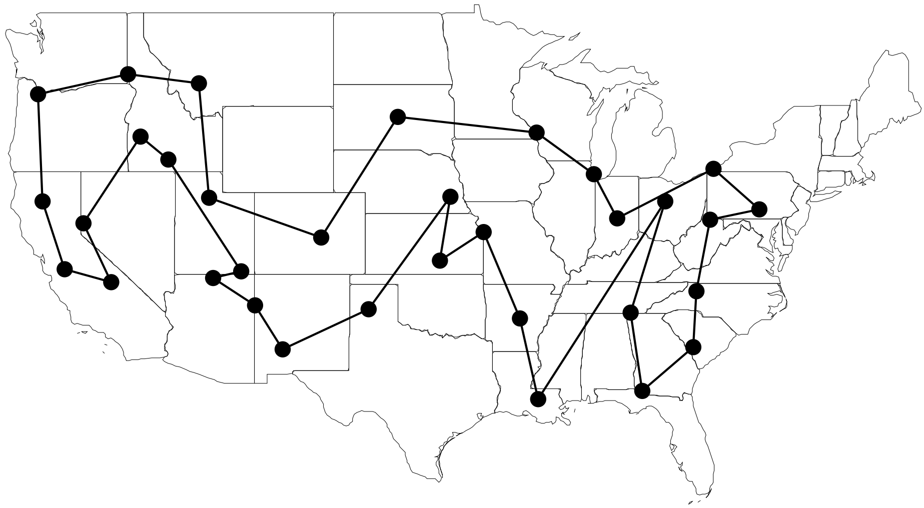


Figure 10.14 Cheapest-insertion solution to *Car 54* TSP instance. Total distance = 12,748 miles.

smallest Δ_{ijk}; see Algorithm 10.3. (In contrast, NI first finds the nearest k to the tour, and then finds the minimum Δ_{ijk} for that k.) CI has a complexity of $O(n^2 \log n)$, slightly higher than NI's $O(n^2)$ complexity. Like NI, CI has a fixed worst-case bound of 2, and this bound is tight (Rosenkrantz et al. 1977). The cheapest-insertion tour for the *Car 54* instance is shown in Figure 10.14 and has a total length of 12,748.

Algorithm 10.3 Cheapest insertion heuristic

choose arbitrary node i_0; $\Theta \leftarrow \{i_0\}$ ▷ Initialization
while $\Theta \neq N$ **do** ▷ Main loop
  if $|\Theta| = 1$ **then**
    $k^* \leftarrow \text{argmin}_{k \in N \setminus \Theta}\{c_{i_0 k}\}$ ▷ Nearest node to i_0
    add (i_0, k^*) and (k^*, i_0) to tour ▷ Insert k^* as second node
  else
    for all $k \in N \setminus \Theta$ and (i, j) on tour **do**
      $\Delta_{ijk} \leftarrow c_{ik} + c_{kj} - c_{ij}$ ▷ Net change due to insertion
    end for
    $k^*, (i^*, j^*) \leftarrow \text{argmin}_{k \in N \setminus \Theta,\ (i,j) \text{ on tour}}\{\Delta_{ijk}\}$ ▷ Best insertion
    remove (i^*, j^*) from tour; add (i^*, k^*) and (k^*, j^*) to tour ▷ Insert k^*
  end if
  $\Theta \leftarrow \Theta \cup \{k^*\}$
end while
return tour

10.4.3 Farthest Insertion

The *farthest insertion* (FI) heuristic is the same as the NI heuristic except that we choose the node k not on the tour that is *farthest* from any node on the tour. We then insert k into

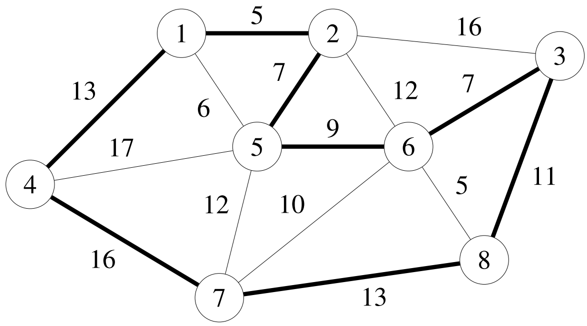


Figure 10.15 Farthest-insertion tour beginning at node 1 for the example instance in Figure 10.5. Total distance = 81.

the edge (i, j) that minimizes Δ_{ijk}, as in NI. In Algorithm 10.2, we simply replace argmin with argmax in line 6. The heuristic has the same complexity as NI: $O(n^2)$.

It may seem counterintuitive to choose farthest nodes to insert, since we are aiming for a shortest tour. But remember that we are only choosing the farthest *node* to insert—we still perform the cheapest *insertion* of that node. The idea is to sketch out the outline of the tour early in the process, and fill in the details later. We will need to visit every node eventually, and there is nothing inherently bad about choosing the far nodes early on.

It is not known whether FI has a fixed worst-case bound. The empirical performance of FI tends to be good, however (Rosenkrantz et al. 1977, Golden and Stewart 1985).

☐ **EXAMPLE 10.4**

Returning to the example in Figure 10.5, let's begin, as usual, at node 1. The farthest node to node 1 is node 3, so we insert it. The farthest node from $\{1, 3\}$ is node 7, at a distance of 17 from node 3. The tour is now $1 \to 7 \to 3$. Node 4 is the farthest nontour node from the tour, and the three possible insertion edges give the following net changes:

$$\begin{aligned} (1,7):& \quad \Delta_{174} = 13 + 16 - 18 = 11 \\ (7,3):& \quad \Delta_{734} = 16 + 33 - 17 = 32 \\ (3,1):& \quad \Delta_{314} = 33 + 13 - 21 = 25 \end{aligned}$$

Therefore, we insert node 4 into the edge $(1, 7)$ to get the tour $1 \to 4 \to 7 \to 3$. Continuing in this way, we make the following insertions:

- Insert 8 into (7,3); $\Delta_{738} = 7$; tour = $1 \to 4 \to 7 \to 8 \to 3$
- Insert 5 into (3,1); $\Delta_{315} = 1$; tour = $1 \to 4 \to 7 \to 8 \to 3 \to 5$
- Insert 2 into (5,1); $\Delta_{512} = 6$; tour = $1 \to 4 \to 7 \to 8 \to 3 \to 5 \to 2$
- Insert 6 into (3,5); $\Delta_{356} = 0$; tour = $1 \to 4 \to 7 \to 8 \to 3 \to 6 \to 5 \to 2$

(See Figure 10.15.) The total distance of this tour is 81.

☐

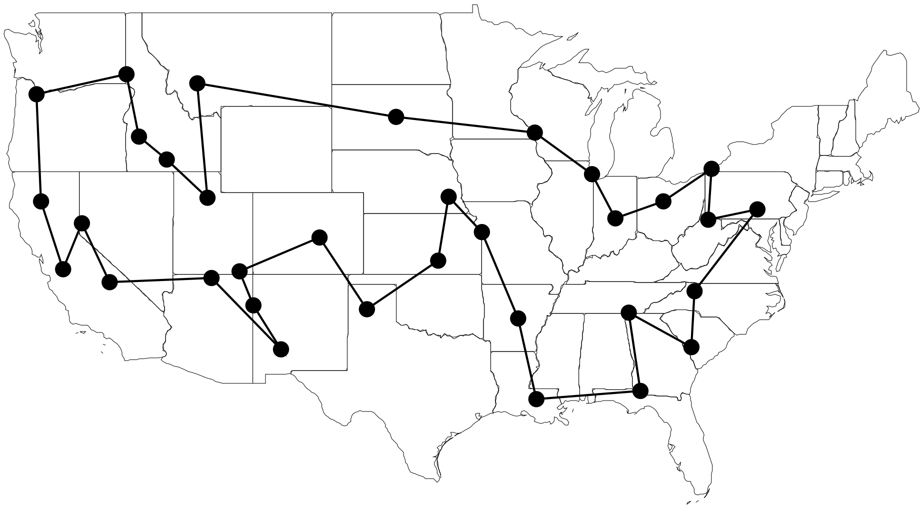


Figure 10.16 Farthest-insertion solution to *Car 54* TSP instance. Total distance = 10,998 miles.

The farthest-insertion tour for the *Car 54* instance, shown in Figure 10.16, has a total distance of 10,998 miles—the best tour of any of our heuristics so far, and only 1.3% worse than the optimal distance of 10,861.

10.4.4 Convex Hull

The *convex hull* heuristic (Stewart 1977) generates an initial partial tour by computing the convex hull of the nodes. Then, at each iteration, the heuristic performs the cheapest insertion, i.e., it finds the nontour node k and the tour edge (i, j) that minimize Δ_{ijk}, and it inserts k into edge (i, j). Like FI, the idea is to generate an outline of the tour quickly and then fill in the remaining nodes. Warburton (1993) proves that the convex hull heuristic has a fixed worst-case bound of 3 for the Euclidean TSP (i.e., the TSP in which distances are Euclidean), though its performance is usually much better than 3 in practice (Golden and Stewart 1985).

□ **EXAMPLE 10.5**

The convex hull of the nodes in the network in Figure 10.5 gives the subtour $1 \to 2 \to 3 \to 8 \to 7 \to 4$. The cheapest insertion is node 6 into edge $(3, 8)$, with $\Delta_{638} = 1$. The only node left is 5, and the cheapest insertion is into edge $(2, 3)$, with $\Delta_{523} = 7$. (Recall that edges not pictured have lengths given by shortest-path distances, so $c_{35} = 16$.) The resulting tour has length 82 and is pictured in Figure 10.17.

(Note that we are treating the nodes as points in Euclidean space to compute the convex hull, but the edge lengths are as given in the figure, not Euclidean distances.)

□

The convex hull heuristic returns a tour of length 11,033 for the *Car 54* instance; see Figure 10.18.

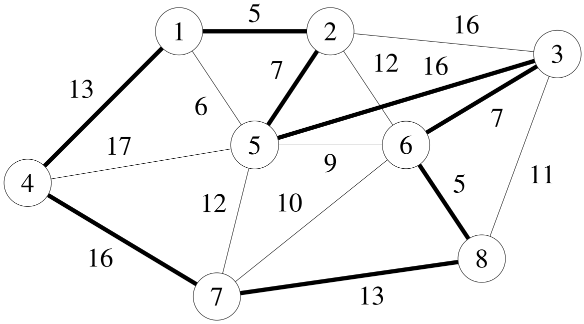


Figure 10.17 Convex-hull tour for the example instance in Figure 10.5. Total distance = 82.

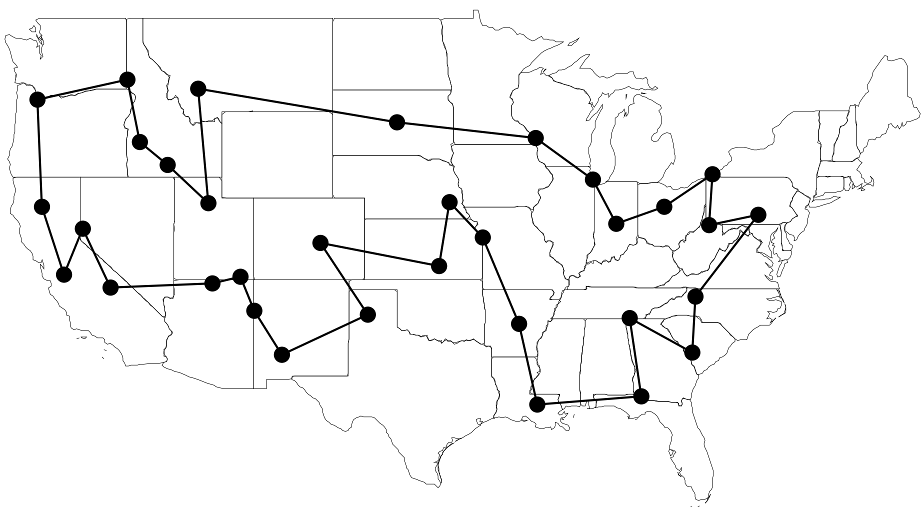


Figure 10.18 Convex-hull solution to *Car 54* TSP instance. Total distance = 11,033 miles.

A variant is the *greatest angle insertion* heuristic (Norback and Love 1977): We begin with the convex hull, and then choose the nontour k and the tour edge (i, j) that maximizes the angle between (i, k) and (k, j), and insert k into this edge.

10.4.5 GENI

All of the insertion heuristics discussed thus far only allow a node to be inserted into an existing edge of the tour. The *generalized insertion* (GENI) heuristic by Gendreau et al. (1992) relaxes this requirement, allowing a node to be inserted between any two nodes on the tour.

Suppose we want to insert a node k between nodes i and j on the tour. To do this, we need to remove one of the edges incident to i and one incident to j and replace them with edges to k. This leaves two nodes—the former neighbors of i and j—that have degree 1 and must be reconnected somehow. (See Figure 10.19(a) and (b). The straight lines represent single edges, while the squiggles represent potentially longer portions of the tour.) The obvious fix is to connect the two former neighbors to each other, as in Figure 10.19(c), but this may be a long edge. Or worse, connecting the two former neighbors may result in two subtours rather than one complete tour, as in Figure 10.19(d). The GENI heuristic tries to find more effective (though also more complicated) ways to reconnect the tour after inserting node k.

GENI considers two types of insertions. Type I insertions remove one additional edge, while Type II insertions remove two additional edges, before reconnecting the tour. We'll use subscripts $+$ and $-$ to denote the successor and predecessor, respectively, of a given node on the tour under a fixed orientation. For example, i^+ is the node that comes after i in the tour.

Let ℓ be a node on the path from j to i for a particular orientation of the tour, with $\ell \neq i, j$. A *Type I* insertion involves

- deleting edges (i, i^+), (j, j^+), and (ℓ, ℓ^+), and
- adding edges (i, k), (k, j), (i^+, ℓ), and (j^+, ℓ^+).

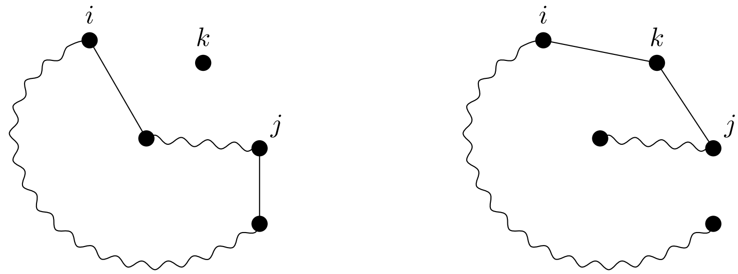
A Type I insertion is depicted in Figure 10.20. In the tour resulting from a Type I insertion, the segments from i^+ to j and from j^+ to ℓ are reversed, while the segment from ℓ^+ to i is unchanged. Note that if $j = i^+$ and $\ell = j^+$, then a Type I insertion is equivalent to the standard insertion—inserting a node between two consecutive nodes.

Let ℓ be a node on the path from j to i and let m be a node on the path from i to j for a particular orientation of the tour, with $\ell \neq j, j^+$ and $m \neq i, i^+$. A *Type II* insertion involves

- deleting edges (i, i^+), (m^-, m), (j, j^+), and (ℓ^-, ℓ), and
- adding edges (i, k), (k, j), (m, j^+), (ℓ^-, m^-), and (i^+, ℓ).

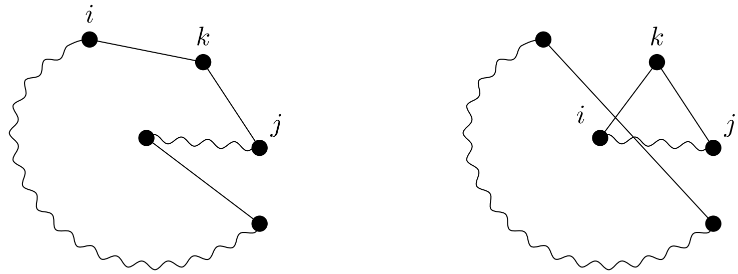
(See Figure 10.21.) Like Type I insertions, Type II insertions result in two tour segments reversing direction, in this case the segments from i^+ to m^- and from m to j.

For a given choice of k, there are $O(n^3)$ choices for (i, j, ℓ) in a Type I insertion and $O(n^4)$ choices for (i, j, ℓ, m) in Type II, times two possible tour orientations. This is a lot of combinations to check, so Gendreau et al. (1992) suggest simplifying the search for good



(a) Tour before insertion.

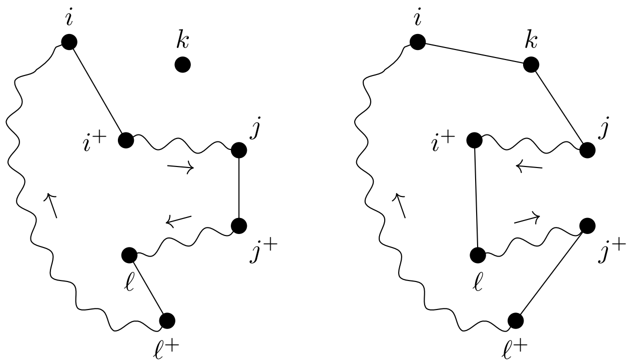
(b) Tour before reconnecting.



(c) Tour after connecting former neighbors of i and j.

(d) Subtours resulting from connecting former neighbors of i and j.

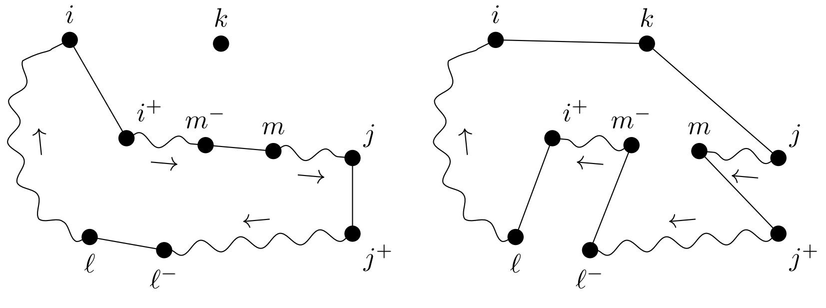
Figure 10.19 Inserting a node between two nonadjacent nodes in the GENI heuristic.



(a) Tour before insertion.

(b) Tour after insertion.

Figure 10.20 GENI Type I insertion.



(a) Tour before insertion.

(b) Tour after insertion.

Figure 10.21 GENI Type II insertion.

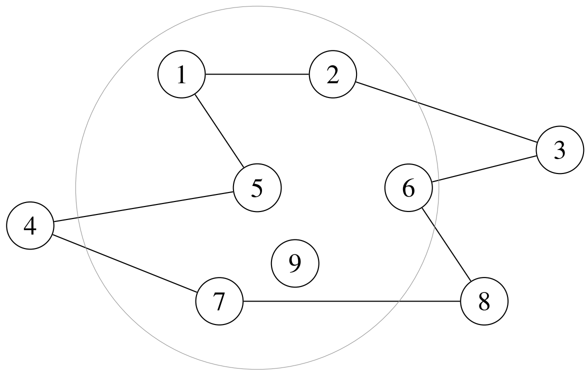


Figure 10.22 4-neighborhood of node 5.

insertions by making use of *p-neighborhoods*. If node i is on the tour, its p-neighborhood $N_p(i)$ is defined as the p nodes on the tour (excluding i itself) that are closest to i as measured by c_{ij}. If the tour has fewer than p nodes in addition to i, then they are all in $N_p(i)$. For example, in the tour in Figure 10.22, $N_4(5) = \{1, 2, 6, 7\}$.

The GENI heuristic restricts the possible combinations for Type I and II insertions as follows:

- $i, j \in N_p(k)$
- $\ell \in N_p(i^+)$
- $m \in N_p(j^+)$ (for Type II).

This significantly reduces the number of combinations to check, while targeting new edges (ℓ, i^+) and (m, j^+) that are short. p is a parameter of the heuristic, chosen by the modeler, and is usually relatively small, say, 5.

When inserting k between consecutive nodes i and j, as in the NI and related heuristics, the total tour length cannot decrease, since $\Delta_{ijk} \geq 0$ in (10.25) due to the triangle inequality. Interestingly, however, GENI insertions do sometimes result in a shorter total tour. (See Problem 10.7.)

The GENI heuristic is summarized in pseudocode in Algorithm 10.4. The heuristic runs in $O(np^4 + n^2)$ time. The heuristic is very effective, especially when combined with the unstringing and stringing (US) improvement heuristic described in Section 10.5.3. GENI insertions have also been incorporated into metaheuristics such as tabu search, variable neighborhood search, and ant colony optimization (Gendreau et al. 1994, Mladenović and Hansen 1997, Le Louarn et al. 2004).

Algorithm 10.4 GENI heuristic

choose 3 arbitrary nodes; construct tour on those nodes ▷ Initialization
construct p-neighborhoods
while tour does not contain all nodes **do** ▷ Main loop
  $k \leftarrow$ random node not on tour
  check all Type I and II insertions for both tour orientations subject to p-neighborhood restrictions
  implement least-cost insertion
  update p-neighborhoods
end while
return tour

10.4.6 Minimum Spanning Tree Heuristic

The next two heuristics we will discuss begin by finding a minimum spanning tree (MST) on the nodes and then converting it to a TSP tour. They therefore function differently from the heuristics described thus far, which iterate through the nodes, inserting one at a time.

Recall that an MST of a network is a minimum-cost tree that contains every node of the network. Finding an MST is easy using a greedy approach. *Prim's algorithm* (Prim 1957) begins with a single node and, at each iteration, adds the shortest edge that connects a node not on the tree to a node on the tree. *Kruskal's algorithm* (Kruskal 1956) begins with an empty tree and, at each iteration, adds the shortest edge (whether or not one of its nodes is contained in the tree) that does not create a cycle. Both algorithms run in $O(|E| \log n)$ time (where E is the set of edges in the network) when implemented using efficient data structures.

Every TSP tour consists of a spanning tree (in particular, a *spanning path*) plus one edge. This suggests we can use MSTs to generate TSP tours, and to derive useful lower bounds.

Lemma 10.9 *Let T^* be an MST on the nodes of a given TSP instance. Then*

$$z(T^*) \leq z^*.$$

Proof. Let P^* be the optimal spanning path through the network and Γ^* be the optimal TSP tour. Remove any edge from Γ^* to obtain a spanning path P; then

$$z(P) \leq z^*$$

by the triangle inequality, and since P^* is the optimal spanning path,

$$z(P^*) \leq z(P).$$

Moreover, since P^* is a spanning tree and T^* is the optimal spanning tree,

$$z(T^*) \leq z(P^*).$$

Combining these inequalities, we get the desired result. ■

The minimum spanning tree heuristic for the TSP begins by finding an MST using Prim's or Kruskal's algorithm or any other method. Next, it doubles every edge to create a network in which every node has even degree. Such a network is called an *Eulerian network* and is guaranteed to have an *Eulerian tour* (a tour that traverses every edge exactly once but may visit each node multiple times):

Theorem 10.10 *An undirected, connected graph has an Eulerian tour if and only if every node has even degree.*

Proof. Omitted; see, e.g., Graver and Watkins (1977). ■

The Eulerian tour derived from the doubled MST gives us a sequence of nodes (each potentially appearing multiple times), which the heuristic converts to a TSP tour simply by skipping any duplicates—this is called *shortcutting*.

The MST heuristic is summarized in Algorithm 10.5.

Algorithm 10.5 MST heuristic

```
1: find an MST
2: double every edge in the MST
3: find an Eulerian tour on the network consisting of the doubled MST
4: construct a TSP tour from the Eulerian tour by shortcutting
5: return tour
```

□ **EXAMPLE 10.6**

Consider again the example in Figure 10.5. An MST is shown in Figure 10.23(a); this tree has a total length of 55. One Eulerian tour is given by $4 \to 1 \to 2 \to 1 \to 5 \to 6 \to 3 \to 6 \to 8 \to 6 \to 7 \to 6 \to 5 \to 1 \to 4$. We convert this to a TSP tour by visiting nodes in the same sequence but skipping duplicates: $4 \to 1 \to 2 \to 5 \to 6 \to 3 \to 8 \to 7$ (Figure 10.23(b)). This tour has a total distance of 81.

The heuristic is very dependent on the Eulerian tour itself. For example, another Eulerian tour on the same MST is $1 \to 5 \to 6 \to 8 \to 6 \to 7 \to 6 \to 3 \to 6 \to 5 \to 1 \to 4 \to 1 \to 2$. The resulting TSP tour is $1 \to 5 \to 6 \to 8 \to 7 \to 3 \to 4 \to 2$, which has a length of 106.

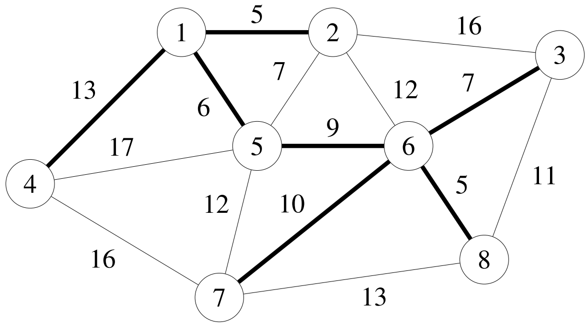
□

The tour returned by the MST heuristic for the *Car 54* instance is shown in Figure 10.24. The tour has length 14,964 miles. Bentley (1992) confirms the relatively poor empirical performance of the MST heuristic.

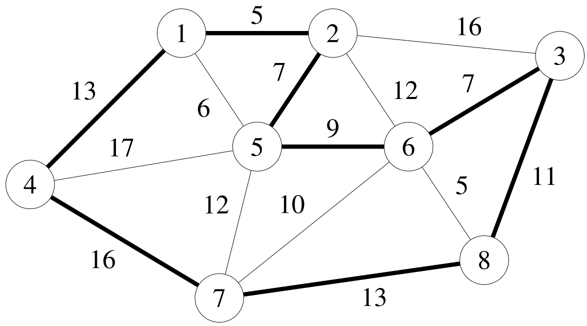
The minimum spanning tree heuristic has a fixed worst-case bound of 2:

Theorem 10.11 *For every instance of the TSP that satisfies the triangle inequality,*

$$\frac{z^{MST}}{z^*} \leq 2, \tag{10.27}$$



(a) Minimum spanning tree.



(b) TSP tour.

Figure 10.23 Tour generated by minimum spanning tree heuristic for instance in Figure 10.5. Total distance = 81.

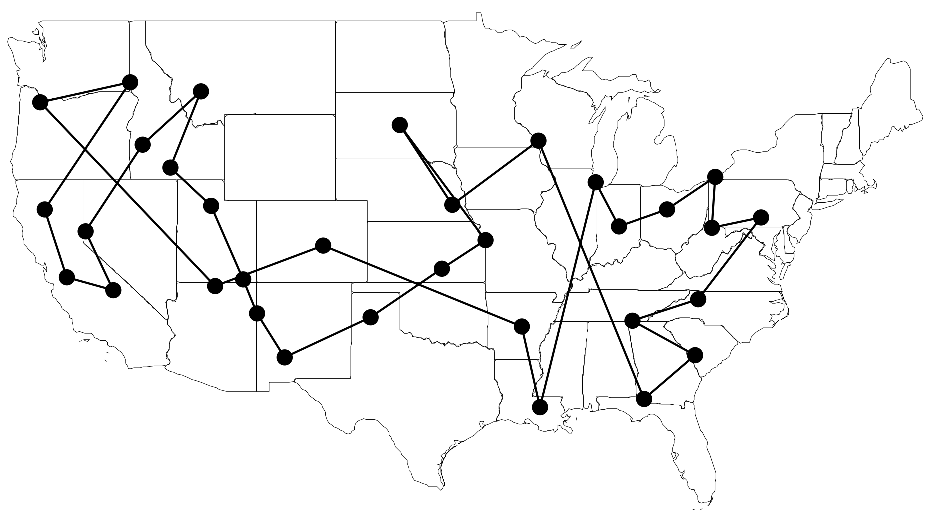


Figure 10.24 Tour obtained by applying MST heuristic to *Car 54* instance. Total distance = 14,964 miles.

where z^* and z^{MST} are the lengths of the optimal tour and the tour returned by the minimum spanning tree heuristic, respectively, for this instance.

Proof. Let T^* and E be the MST and the Eulerian tour found by the heuristic. Since E traverses every edge in T^* exactly twice,

$$z(E) = 2z(T^*) \leq 2z^*,$$

where the inequality is by Lemma 10.9. By the triangle inequality,

$$z^{MST} \leq z(E),$$

since the minimum spanning tree heuristic replaces edge pairs of the form (i, k), (k, j) with edges of the form (i, j). ■

This bound is tight; see Problem 10.12.

10.4.7 Christofides' Heuristic

Like the minimum spanning tree heuristic, Christofides' heuristic (Christofides 1976) builds an MST and then converts it to a TSP tour. The difference is in the way the heuristics modify the MST to create an Eulerian network. Whereas the minimum spanning tree heuristic simply doubles every edge, Christofides' heuristic takes a more sophisticated, and more effective, approach. By Theorem 10.10, it is the odd-degree nodes that prevent the MST from having an Eulerian tour. Christofides's heuristic adds a single edge to each of these nodes to produce a graph whose nodes all have even degree. It is possible to pair up the odd-degree nodes because there are always an even number of them:

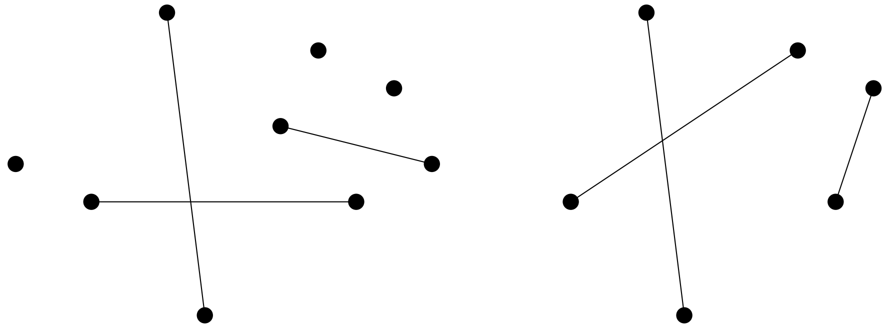
Lemma 10.12 (Handshaking Lemma) *In any undirected graph, the number of odd-degree nodes is even.*

Proof. Omitted; see Problem 10.8. ■

(This lemma is called the *handshaking lemma*: If a set of people shake hands at a party, there must be an even number of people who shook hands with an odd number of people.)

A *matching* on a set of nodes is a set of edges such that every node is contained in at most one edge. A *perfect matching* is a set of edges such that every node is contained in exactly one edge. (See Figure 10.25.) Christofides' heuristic finds a *minimum-weight perfect matching* on the odd-degree nodes in the MST, where the weights are given by the edge lengths. A minimum-weight perfect matching can be found in polynomial time (Edmonds 1965). The matching may duplicate edges already in the MST. A perfect matching on the odd-degree nodes must exist, by Lemma 10.12. When the matching edges are added to the MST, every node has even degree (since we are adding a single edge to each odd-degree node), so the resulting network is Eulerian. The heuristic then finds an Eulerian tour and converts it to a TSP tour by shortcutting, exactly as in the minimum spanning tree heuristic. The heuristic is summarized in Algorithm 10.6.

☐ **EXAMPLE 10.7**



(a) A matching. (b) A perfect matching.

Figure 10.25 Matchings.

Algorithm 10.6 Christofides' heuristic

find an MST
find a minimum-weight perfect matching on the odd-degree nodes in the MST
find an Eulerian tour on the network consisting of the MST plus the matching edges
construct a TSP tour from the Eulerian tour by shortcutting
return tour

Figure 10.23(a) in Example 10.6 shows an optimal spanning tree for the network in Figure 10.5. In this MST, nodes 1, 2, 3, 4, 7, and 8 have odd degree. The optimal matching consists of the edges $(1,2)$, $(3,8)$, and $(4,7)$. (See Figure 10.26.) One Eulerian tour on the edges from the MST and the matching is: $4 \rightarrow 1 \rightarrow 2 \rightarrow 1 \rightarrow 5 \rightarrow 6 \rightarrow 3 \rightarrow 8 \rightarrow 6 \rightarrow 7 \rightarrow 4$. Shortcutting, we get the following TSP tour: $4 \rightarrow 1 \rightarrow 2 \rightarrow 5 \rightarrow 6 \rightarrow 3 \rightarrow 8 \rightarrow 7$ (Figure 10.26(b)). This tour has a total distance of 81.

□

Christofides' heuristic produces an 11,654-mile tour for the *Car 54* instance (Figure 10.27), much better than the 14,964-mile tour returned by the MST heuristic (Figure 10.24) and approximately 7.3% longer than the optimal tour.

Christofides' heuristic has the best fixed worst-case bound known to date.

Theorem 10.13 *For every instance of the TSP that satisfies the triangle inequality,*

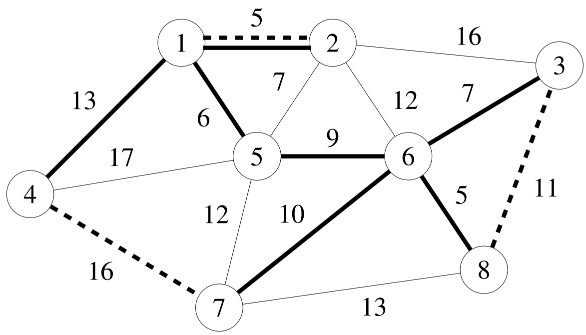
$$\frac{z^{CH}}{z^*} \le \frac{3}{2}, \tag{10.28}$$

where z^* and z^{CH} are the lengths of the optimal tour and the tour returned by Christofides' heuristic, respectively, for this instance.

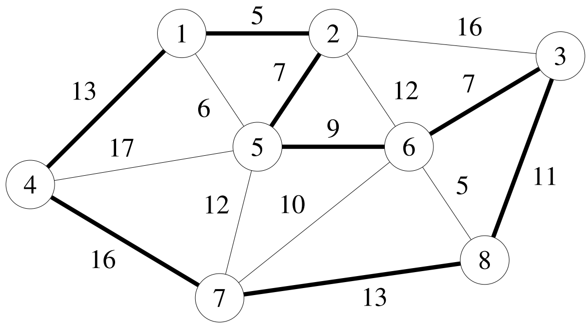
Proof. Let Γ' be an optimal TSP tour on the odd-degree nodes in T^* (the MST found by the heuristic). Clearly

$$z(\Gamma') \le z^* \tag{10.29}$$

by the triangle inequality, since Γ' is the optimal TSP tour for a subset of N. Γ' consists of two disjoint perfect matchings on the odd-degree nodes, each consisting of alternating



(a) Minimum spanning tree (solid lines) and optimal matching (dashed lines).



(b) TSP tour.

Figure 10.26 Tour generated by Christofides' heuristic for instance in Figure 10.5. Total distance = 81.

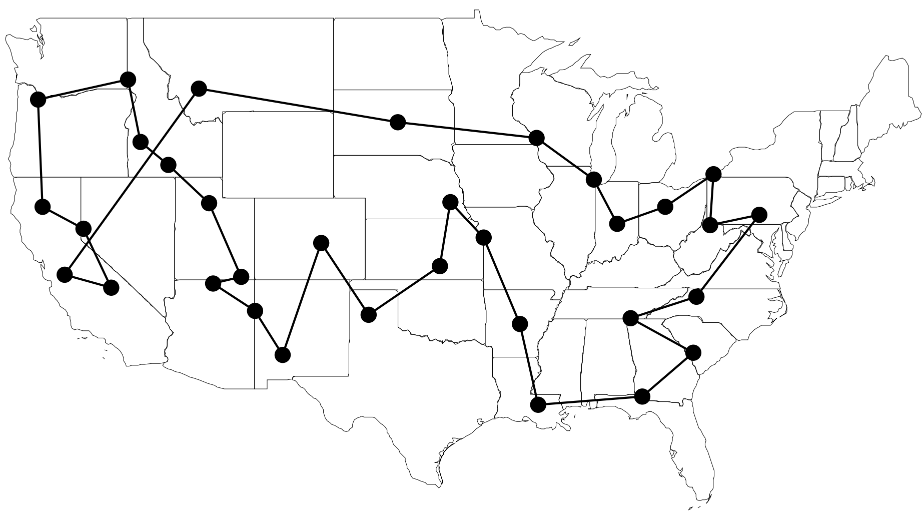


Figure 10.27 Tour obtained by applying Christofides' heuristic to *Car 54* instance. Total distance = 11,654 miles.

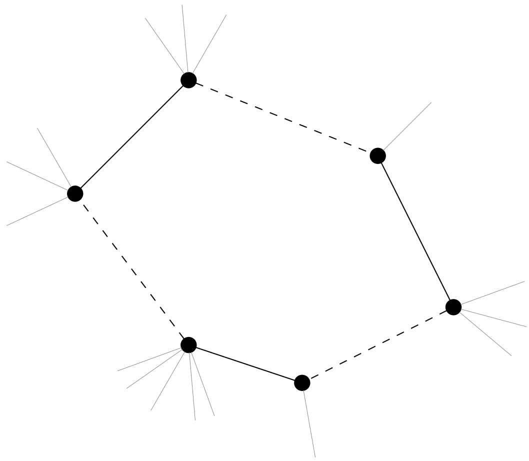


Figure 10.28 Optimal TSP tour on odd-degree nodes in MST in Christofides' heuristic, and two perfect matchings (solid and dashed) that constitute it.

edges in the tour. (See Figure 10.28.) Let M' be the shorter of these two matchings. Clearly

$$z(M') \le \frac{z(\Gamma')}{2}.$$

Let M^* be the optimal perfect matching on the odd-degree nodes. Since $z(M^*) \le z(M')$, we have

$$z(M^*) \le \frac{z(\Gamma')}{2} \le \frac{z^*}{2}. \tag{10.30}$$

Combining Lemma 10.9 and (10.30), we have

$$z(T^*) + z(M^*) \le \frac{3}{2}z^*.$$

The TSP tour returned by the heuristic is obtained from $T^* \cup M^*$ by shortcutting; by the triangle inequality,

$$z^{CH} \le z(T^*) + z(M^*).$$

■

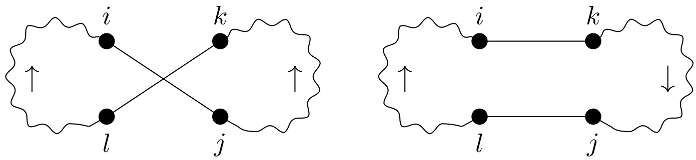
The $3/2$ bound in Theorem 10.13 is tight; see Problem 10.13.

10.5 IMPROVEMENT HEURISTICS FOR THE TSP

The tours returned by the construction heuristics discussed in Section 10.4 are typically not optimal. In this section, we discuss improvement heuristics for the TSP that begin with a complete tour and perform operations on it to try to make it shorter.

10.5.1 k-Opt Exchanges

One can tell just by looking at Figure 10.10 that the nearest-neighbor solution for the *Car 54* instance is bad. The most obvious red flag is that the tour crosses itself several times.



(a) Replacing with (i, j) and (k, ℓ). (b) Replacing with (i, k) and (j, ℓ).

Figure 10.29 Two possible ways to reconnect tour during 2-opt exchange.

Lemma 10.14 *If the nodes each have coordinates in $\mathbb{R}^2$ and the distance c_{ij} is equal to the Euclidean distance between nodes i and j, then the optimal TSP tour does not cross itself.*

Proof. Omitted; see Problem 10.9. ■

Of course, the *Car 54* distances are not Euclidean (they come from road networks), but you might still suspect that we can improve the tour in Figure 10.10 by "uncrossing" it. One mechanism for doing this is called a *2-opt* exchange. In a 2-opt, we remove two edges and replace them with two other edges to form a new tour. For a given pair of edges, there are two possible ways to reconnect the tour, shown in Figure 10.29. However, the replacement edges in Figure 10.29(a) are the same as the original edges that were removed, so the only real option is the strategy in Figure 10.29(b). (Replacing (i, j) and (k, ℓ) with (i, ℓ) and (j, k) is also not an option since this creates two subtours rather than one contiguous tour.) The 2-opt idea was introduced by Flood (1956) (in the very first issue of the journal *Operations Research*, after it subsumed the *Journal of the ORSA*) as a way to fix crossing tours generated by his algorithm; shortly thereafter, Croes (1958) developed a construction heuristic with 2-opt exchanges as the centerpiece.

To describe the 2-opt exchange more formally, suppose (i, j) and (k, ℓ) are disjoint edges in the tour and that, for a given orientation of the tour, node i comes before j and node k comes before ℓ. Then a 2-opt exchange consists of replacing edges (i, j) and (k, ℓ) with edges (i, k) and (j, ℓ). This also changes the orientation of the $j \to \cdots \to k$ portion of the tour. For Euclidean problems, a given 2-opt exchange will only reduce the total length of the tour if edges (i, j) and (k, ℓ) cross each other, since otherwise the new edges (i, j) and (k, ℓ) will cross each other.

The 2-opt heuristic iterates through all pairs of edges looking for 2-opt exchanges that improve the objective function. The procedure terminates when no such pairs can be found. At each iteration, there are $\binom{n}{2} \sim O(n^2)$ pairs of edges to consider. The heuristic is simple and effective. A single 2-opt exchange (exchanging edges $(6, 8)$ and $(3, 7)$ for edges $(6, 3)$ and $(8, 7)$) is sufficient to "fix" the nearest-neighbor tour for the 8-node problem in Example 10.2. When applied to the nearest-neighbor *Car 54* solution in Figure 10.10, the 2-opt heuristic results in the tour shown in Figure 10.30, whose length is 10,944 miles—only 0.76% longer than the optimal tour.

Building on this idea, we can try removing three edges and replacing them with alternative edges. This is the idea behind the *3-opt* exchange (Lin 1965). When we remove three disjoint edges, there are eight possible ways to reconnect the tour, as shown in Figure 10.31.

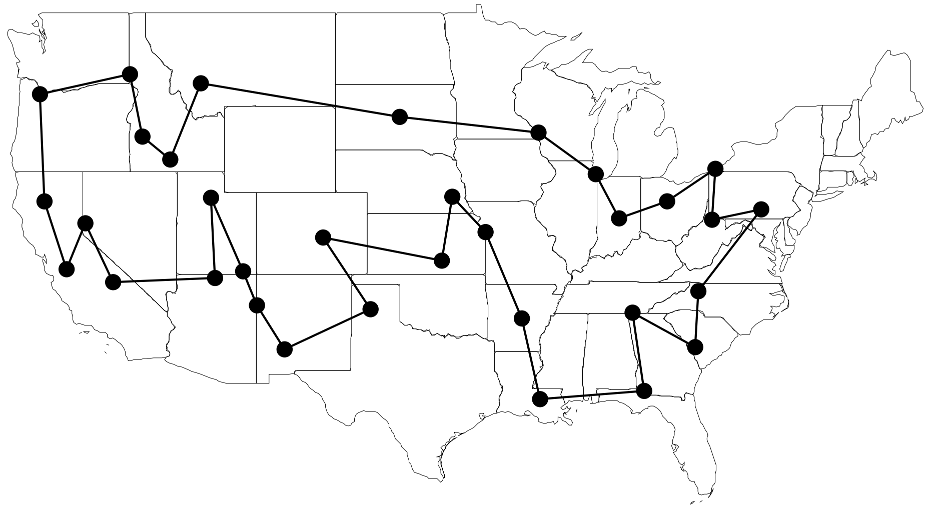


Figure 10.30 Tour obtained by applying 2-opt heuristic to nearest-neighbor tour for *Car 54* instance. Total distance = 10,944 miles.

The first reconnection in Figure 10.31 is identical to the original tour, but the remaining seven strategies could all potentially result in shorter tours. The 3-opt method is powerful but also computationally costly: There are $\binom{n}{3} \sim O(n^3)$ triplets of edges to consider, and for each, we must evaluate seven possible reconnections. (Recall that 2-opt requires considering $O(n^2)$ pairs of edges and evaluating one possible reconnection for each.) We could even try general k-opt exchanges for $k = 4, 5, \ldots$, but the computational burden makes the search for such exchanges impractical.

The search for higher-order k-opt exchanges is not hopeless, however. The *Lin–Kernighan heuristic* (Lin and Kernighan 1973) finds k-opt exchanges by aggregating multiple 2-opt exchanges, allowing some intermediate exchanges that may even increase the tour length so long as the ultimate result is an improvement. Johnson and McGeoch (1997) report computational experiments showing the Lin–Kernighan heuristic to come within 1.5% of the optimal tour, on average, for random instances with up to a million nodes. The heuristic is so powerful that it is a component of nearly every modern exact algorithm for the TSP, even though the heuristic itself is over 35 years old (Applegate et al. 2007). An important enhancement is the *chained Lin–Kernighan heuristic* by Martin et al. (1991), which contains a feature for the algorithm to "kick" the tour, i.e., to modify it slightly, when the search appears to be stuck. The variant on chained Lin–Kernighan by Johnson (1995) can solve large instances to within 0.1% optimality (Johnson and McGeoch 1997).

10.5.2 Or-Opt Exchanges

An *Or-opt* exchange (Or 1976) takes a segment of the tour consisting of p consecutive nodes and moves it to another spot on the tour, possibly reversing the order of the p nodes as well. A few examples with $p = 3$ are shown in Figure 10.32. Typically, p is relatively small; one common implementation searches for exchanges with $p = 3$, then repeats for $p = 2$, then for $p = 1$, and then terminates. Or-opt can be implemented efficiently, with $O(n^2)$ complexity at each iteration.

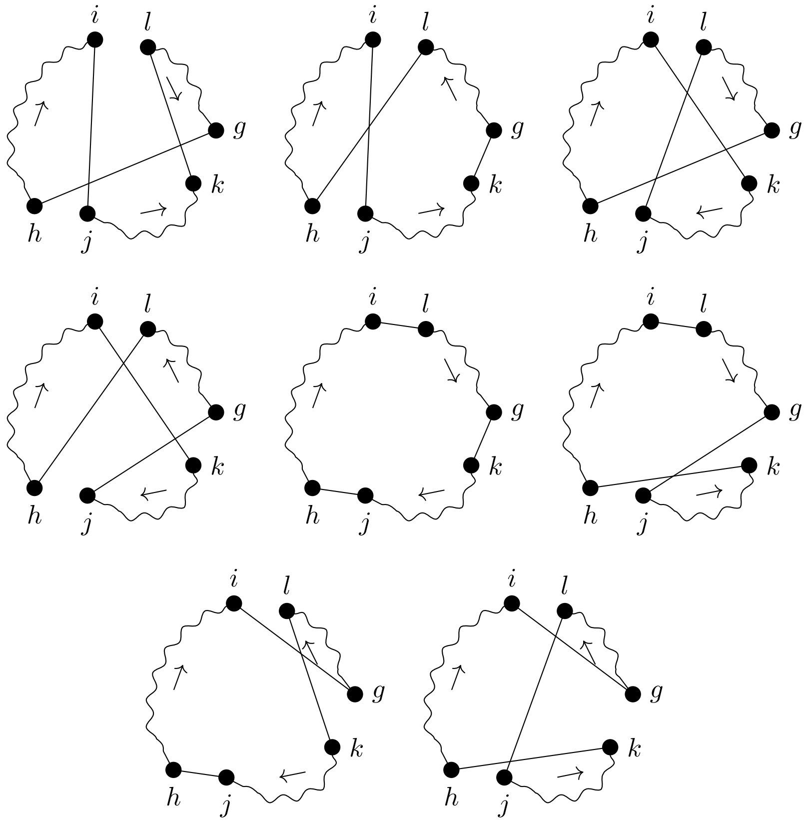


Figure 10.31 Eight possible ways to reconnect tour during 3-opt exchange.

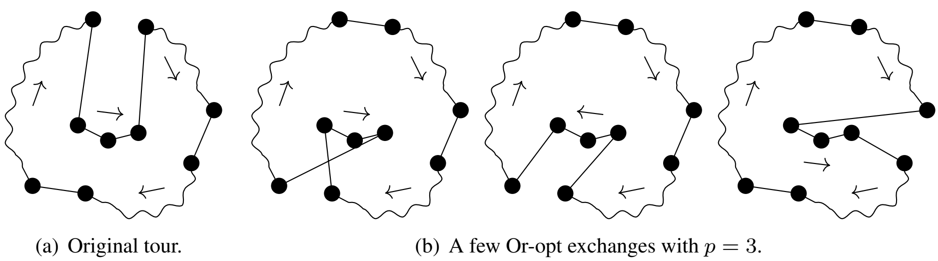


Figure 10.32 Or-opt exchanges.

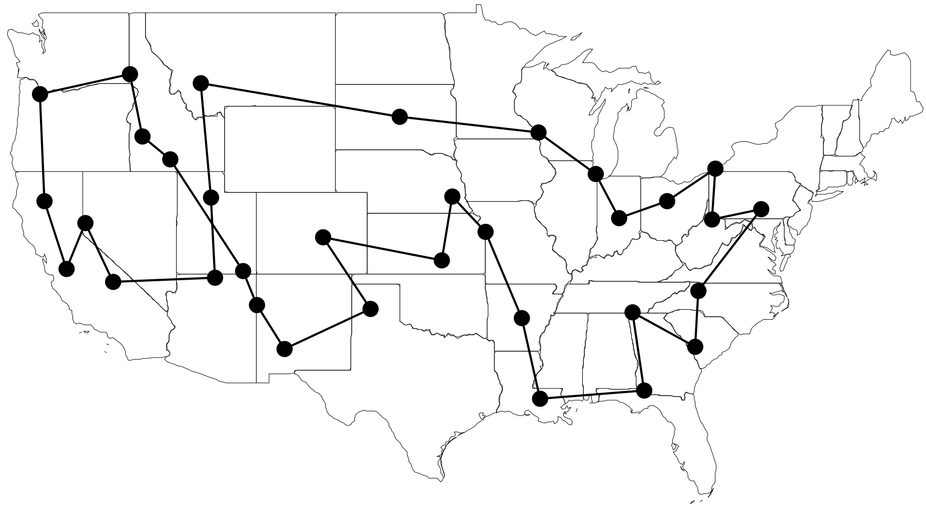


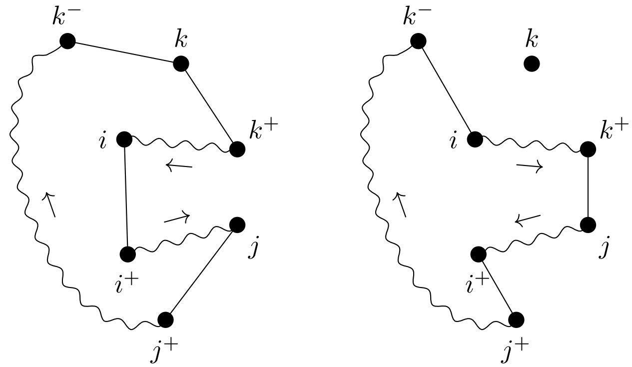
Figure 10.33 Tour obtained by applying Or-opt heuristic to nearest-neighbor tour for *Car 54* instance. Total distance = 11,212 miles.

A single Or-opt exchange—moving node 3 so that it comes after node 6 rather than node 8 (orienting the tour clockwise)—is sufficient to "fix" the nearest-neighbor tour in Example 10.2. The iterative approach described above, applied to the *Car 54* nearest-neighbor tour (Figure 10.30), performs one Or-opt with $p = 3$, three with $p = 2$, and four with $p = 1$. The resulting tour is shown in Figure 10.33 and has a total length of 11,212 miles—not quite as impressive as the 10,944-mile tour found by 2-opt, but close. Actually, a few 2-opt exchanges applied after the Or-opt procedure would convert the tour to the 10,944-mile solution. The idea of combining Or-opt and 2-opt exchanges is actually quite powerful, with the ability to generate high-quality tours while being much simpler to code than Lin–Kernighan and other hybrid exchange heuristics (Babin et al. 2007).

10.5.3 Unstringing and Stringing

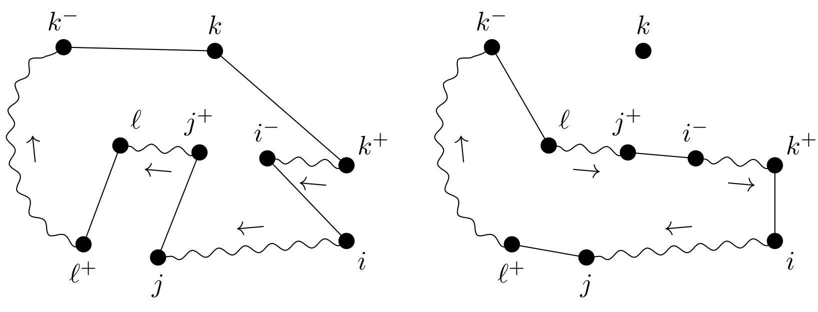
Gendreau et al. (1992) propose an improvement heuristic called *unstringing and stringing* (US). The idea is to remove a node from the tour and then reinsert it at another spot. (Think of removing a bead from a string and then replacing it elsewhere on the string.) This sounds like an Or-opt exchange with $S = 1$, but the difference is that the tour need not simply close up around the removed node, and the insertion need not occur between two consecutive nodes.

In the US heuristic, "stringing" occurs using Type I or Type II GENI insertions (see Section 10.4.5), while "unstringing" is the reverse. In particular, suppose that the path from k^+ to k^- contains nodes called i, j, and ℓ, in that order. (Recall that the superscripts $+$ and $-$ refer to successors and predecessors.) Then a Type I unstringing of node k removes edges (k^-, k), (k, k^+), (i, i^+), and (j, j^+) and replaces them with (k^-, i), (k^+, j), and (i^+, j^+); see Figure 10.34. A Type II unstringing of node k removes edges (k^-, k), (k, k^+), (i^-, i), (j, j^+), and (ℓ, ℓ^+) and replaces them with (k^-, ℓ), (j^+, i^-), (k^+, i), and (j, ℓ^+); see Figure 10.35. As in the GENI heuristic, we restrict the possible combinations by making use of p-neighborhoods; see Gendreau et al. (1992) for details.



(a) Tour before unstringing. (b) Tour after unstringing.

Figure 10.34 US Type I unstringing of node k.



(a) Tour before unstringing. (b) Tour after unstringing.

Figure 10.35 US Type II unstringing of node k.

Gendreau et al. (1992) recommend combining the GENI construction heuristic and the US improvement heuristic to obtain—naturally—the GENIUS heuristic. They report favorable computational results for the GENIUS heuristic compared to several others, including 2-opt and Lin–Kernighan (though on a rather limited set of instances).

10.6 BOUNDS AND APPROXIMATIONS FOR THE TSP

Using several heuristics, we have found a tour with total distance 81 for the instance in Figure 10.5. This might suggest that this tour is optimal—but how can we know for sure? If we did not want to or could not solve the instance using an exact method, an alternative is to find a lower bound on the optimal tour length. If the lower bound equals 81, then we know our tour is optimal. If not, then at least we have a benchmark against which to compare our solution.

Theorems 10.1 and 10.11 provide lower bounds on the optimal TSP tour length. For the instance in Figure 10.5, these theorems provide bounds of 79 and 55, respectively. In Sections 10.6.1 and 10.6.2, we discuss two more powerful and flexible lower bounds. We also discuss the related question of bounding the integrality gap, i.e., the gap between the TSP and its LP relaxation, in Section 10.6.3. In Section 10.6.4, we discuss approximation bounds, i.e., bounds on the constant η in (10.21) that can be proven for a polynomial-time heuristic. Then, in Section 10.6.5, we discuss the asymptotic behavior of the optimal tour length as $n \to \infty$, including the well-known *square-root* approximation for the TSP.

10.6.1 The Held–Karp Bound

From Lemma 10.9, we know that the total length of the edges in an MST is a lower bound on that of the optimal TSP tour. This is an interesting fact, but not usually a particularly strong bound. For example, an MST for the instance in Figure 10.9 has total length 55 (much smaller than our current best solution of 81). Similarly, an MST for the *Car 54* instance has total length 9425; it is pictured in Figure 10.36. This bound is 13.2% smaller than the optimal tour length of 10,861.

One obvious problem with the MST bound is that a spanning tree only has $n - 1$ edges, whereas a TSP tour has n. This suggests we can improve the bound by adding another edge, and indeed, this is the idea behind one of the most important lower bounds that have been developed, called the *Held–Karp bound* (Held and Karp 1970). The bound is derived from a construct that Held and Karp call a *1-tree*, defined as a spanning tree on nodes $\{2, \ldots, n\}$ plus two edges incident to node 1.[2]

Every TSP tour is a 1-tree, and therefore the problem of finding the minimum-weight 1-tree is a relaxation of the TSP.

Lemma 10.15 *Let $\hat{T}^*$ be an optimal 1-tree for a given TSP instance. Then*

$$z(\hat{T}^*) \le z^*.$$

[2]The name 1-tree refers to the special treatment of node 1. It is sometimes casually interpreted as referring to a spanning tree plus 1 edge, but this definition is too loose—it includes structures that are not 1-trees.

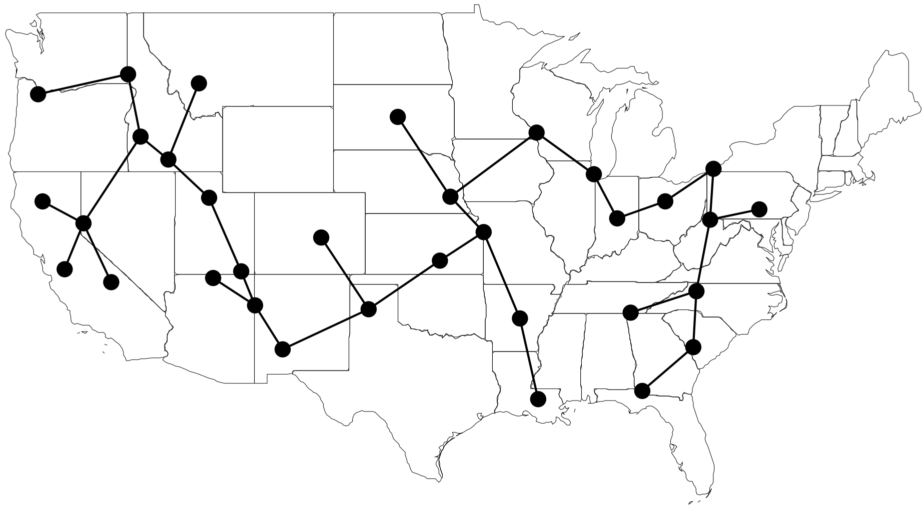


Figure 10.36 Minimum spanning tree for *Car 54* instance. Total distance = 9425 miles.

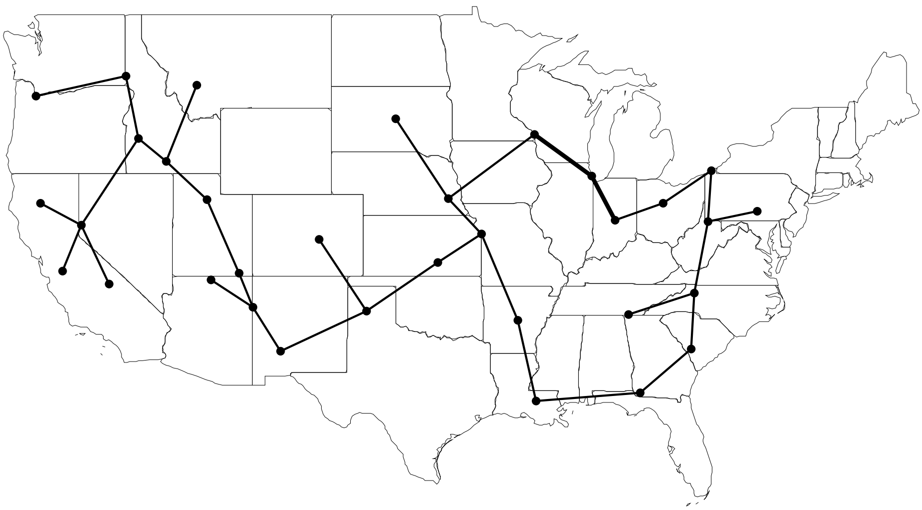
Moreover, it is easy to find an optimal 1-tree for a given instance: We simply find an MST on $\{2, \ldots, n\}$ and then add the two shortest edges connecting node 1 to the MST. Figure 10.37(a) shows the optimal 1-tree for the *Car 54* instance. It has a total length of 9866 miles, providing a better bound on the optimal TSP tour than the MST. We can get even tighter bounds by building optimal 1-trees rooted at each node (that is, labeling each node "node 1") in turn; the longest such 1-tree is rooted at Manuelito, TX and has a total distance of 10,007 miles (Figure 10.37(b)).

But we can do even better. To see how, consider the network in Figure 10.38(a). The labels on the edges indicate their lengths. Edges that are not included in the figure are assumed to have a distance of ∞. The optimal TSP length is 20 units, for example, for the tour $1 \to 3 \to 2 \to 6 \to 4 \to 5$. An optimal 1-tree on this network, shown in Figure 10.38(b), has a total length of 10. So far, this is not too impressive a bound.

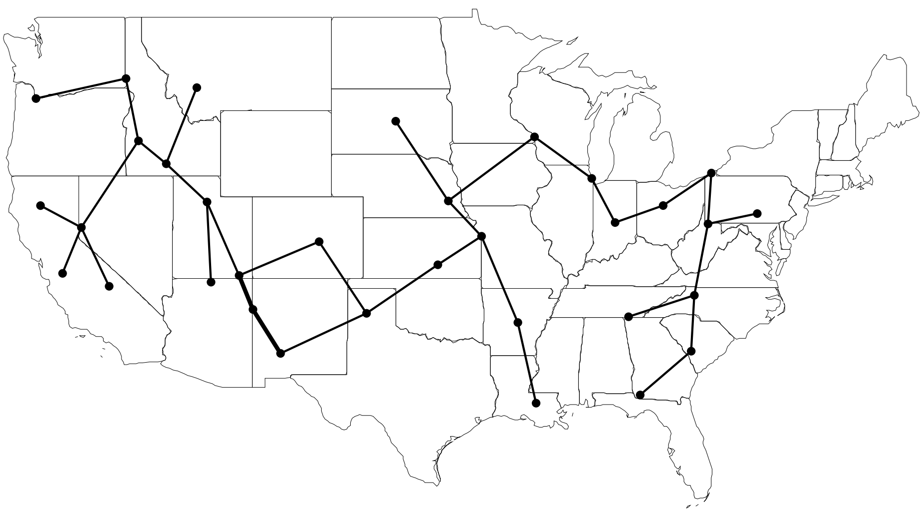
Now suppose we increase the lengths of all edges incident to node 4 by 10 units (Figure 10.38(c)). This increases the length of any TSP tour by exactly 20 units, since every TSP tour traverses exactly two of these edges. Therefore, the optimal TSP tour doesn't change, though its length does, to 40 units. The optimal 1-tree, shown in Figure 10.38(d), now has length 40. By Lemma 10.15, 40 is a lower bound on the optimal TSP tour for the revised network, and since we have a tour of length 40, that tour must be optimal. Moreover, since the optimal tour for the revised network is also optimal for the original network, the 1-tree on the revised network provides us a guarantee that our original TSP tour is optimal.

This idea becomes even more powerful when we add weights (constants) to the edges incident to multiple nodes—possibly all of them. Suppose we add a weight of λ_i to all of the edges incident to node i. (We'll usually just say "add a weight of λ_i to node i.") Let $\lambda = (\lambda_1, \ldots, \lambda_n)$ be the vector of weights. This gives us a revised distance matrix c'_{ij}:

$$c'_{ij} = c_{ij} + \lambda_i + \lambda_j. \tag{10.31}$$

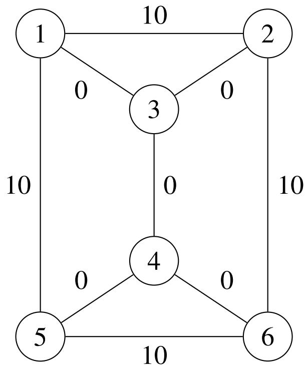


(a) Optimal 1-tree rooted at Chicago. Total distance = 9866 miles.

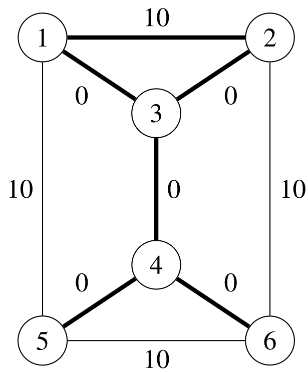


(b) Optimal 1-tree rooted at Manuelito. Total distance = 10,007 miles.

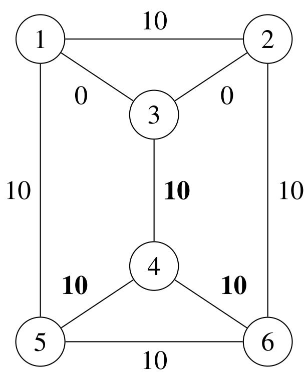
Figure 10.37 Optimal 1-trees for *Car 54* instance.



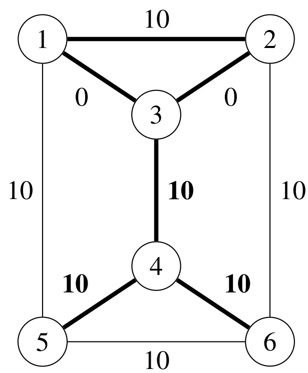
(a) Original network.



(b) Optimal 1-tree on original network.



(c) Revised network.



(d) Optimal 1-tree on revised network.

Figure 10.38 1-tree example.

We'll use $z(\cdot)$ and $z'(\cdot)$ to refer to total distances under the original and revised distance matrices, respectively.

Lemma 10.16 *Let $\lambda \in \mathbb{R}^n$.*

(a) If Γ is a TSP tour, then

$$z'(\Gamma) = z(\Gamma) + 2\sum_{i \in N} \lambda_i.$$

(b) For any $\lambda \in \mathbb{R}^n$, if Γ^* is an optimal TSP tour under the distance matrix c, then it is also optimal under the distance matrix c'.

(c) For any 1-tree $\hat{T}$,

$$z'(\hat{T}) = z(\hat{T}) + \sum_{i \in N} d_i(\hat{T})\lambda_i, \tag{10.32}$$

where $d_i(\hat{T})$ is the degree of node i in 1-tree $\hat{T}$.

Proof. Omitted; see Problem 10.15. ■

On the other hand, changing the distance matrix can change not only the *weight* of the optimal 1-tree but also the structure of the 1-tree itself. As the weights λ change, the length of the optimal TSP tour changes in sync, but the length of the optimal 1-tree may "jump," providing better bounds.

Theorem 10.17 *For any $\lambda \in \mathbb{R}^n$,*

$$z^* \geq z(\hat{T}^*) + \sum_{i \in N} \lambda_i \left(d_i(\hat{T}^*) - 2\right), \tag{10.33}$$

where $\hat{T}^*$ is the optimal 1-tree under the revised distance matrix given by (10.31).

Proof. By Lemmas 10.15 and 10.16,

$$\begin{aligned} & z'(\Gamma^*) \geq z'(\hat{T}^*) \\ \Longrightarrow\ & z(\Gamma^*) + 2\sum_{i \in N} \lambda_i \geq z(\hat{T}^*) + \sum_{i \in N} d_i(\hat{T}^*)\lambda_i \\ \Longrightarrow\ & z^* = z(\Gamma^*) \geq z(\hat{T}^*) + \sum_{i \in N} \lambda_i \left(d_i(\hat{T}^*) - 2\right), \end{aligned}$$

where Γ^* is the optimal TSP tour. ■

The right-hand side of (10.33) is the *Held–Karp lower bound* and is denoted $z^{HK}(\lambda)$. This bound holds for any λ; now the question is how to find good values of λ so that the bound is as tight as possible. Held and Karp (1971) propose using subgradient optimization to improve λ. We start with an arbitrary vector λ^0, and at each iteration t, after calculating the lower bound, we update λ as follows:

$$\lambda_i^{t+1} = \lambda_i^t + \Delta^t \left(d_i^t - 2\right), \tag{10.34}$$

where

$$\Delta^t = \frac{\alpha^t(\mathrm{UB} - z^{HK}(\lambda^t))}{\sum_{i \in N}(d_i^t - 2)^2}, \tag{10.35}$$

d_i^t is the degree of node i in the 1-tree found during iteration t (i.e., the optimal 1-tree for weights λ^t), and UB is the best upper bound found so far. As usual, α^t is a constant that is halved when a given number of consecutive iterations passes without improving the lower bound (see page 673).

A set of weights λ tends to provide a better bound if the resulting 1-tree "looks like" a TSP tour, i.e., its nodes mostly have degree 2. (Why?) One way to think about the update step (10.34)–(10.35) is that nodes that have degree 1 receive lower weights in the next iteration (encouraging more edges to be incident to them in the next 1-tree) and nodes that have degree greater than 2 receive higher weights (encouraging fewer edges).

How tight can the bound be? Held and Karp prove that if we could find the optimal λ, the resulting bound would equal the LP relaxation bound:

$$z^{HK}(\lambda^*) = z^{LP}. \tag{10.36}$$

(For this reason, the name "Held–Karp lower bound" is often used to refer to the LP relaxation bound itself.) In practice we can't usually find the optimal λ, but we can find good enough λ that the Held–Karp bound is very close to the LP bound—within 0.01% in computational experiments by Johnson and McGeoch (1997).

If all of this is starting to feel a lot like Lagrangian relaxation, you're not wrong. Held and Karp (1970) show that their lower bound is equivalent to the Lagrangian relaxation bound obtained from the formulation (10.7)–(10.10). The constraints in the Lagrangian subproblem define a polyhedron whose vertices are the set of all 1-trees. The subproblem therefore has the integrality property, which explains (10.36). Held and Karp (1971) propose subgradient optimization to update the multipliers and show how to embed this approach into a branch-and-bound scheme to find feasible—often optimal—solutions in addition to the lower bounds. (See also Balas and Toth (1985).)

☐ **EXAMPLE 10.8**

Let $\lambda_5 = 10$ and $\lambda_i = 0$ for the other nodes i in Figure 10.5. Figure 10.39 shows the revised edge lengths and the resulting optimal 1-tree. The right-hand side of (10.33) for this 1-tree is 65. This is a lower bound to compare against our best known solution, from Example 10.4 and others, whose length is 81, but it is quite a weak lower bound.

An optimal bound results from setting $\lambda = (10, 10, 1, 0, 10, 7, 5, 3)$. The optimal 1-tree, pictured in Figure 10.40, gives a lower bound of 81, proving (finally!) that the tour in Example 10.4 is optimal. In fact, the 1-tree in Figure 10.40 *is* the tour in Example 10.4.

☐

In Example 10.8, the 1-tree problem gave a feasible solution to the TSP, even though it is a relaxation. This does not happen for every instance, but it is not uncommon. For example, for the *Car 54* instance, a few hundred iterations of the subgradient optimization procedure described above yield a vector λ whose optimal 1-tree is the optimal TSP tour in Figure 10.3, with length 10,861.

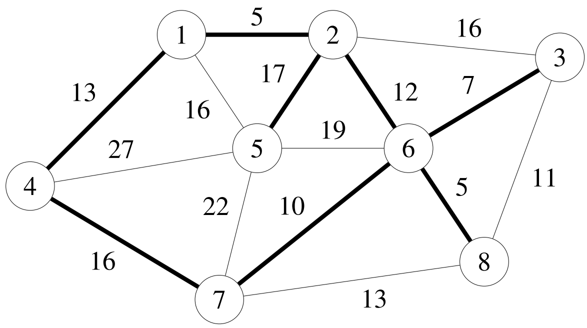


Figure 10.39 Optimal 1-tree for instance in Figure 10.5 with $\lambda_5 = 10$. Resulting lower bound = 65.

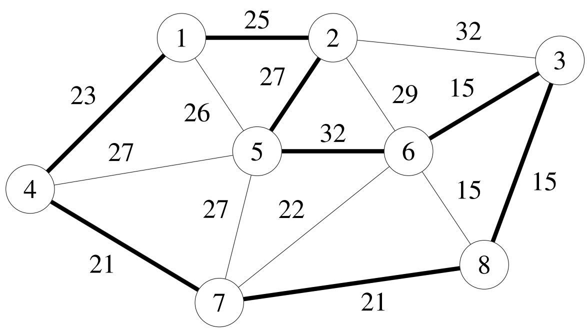
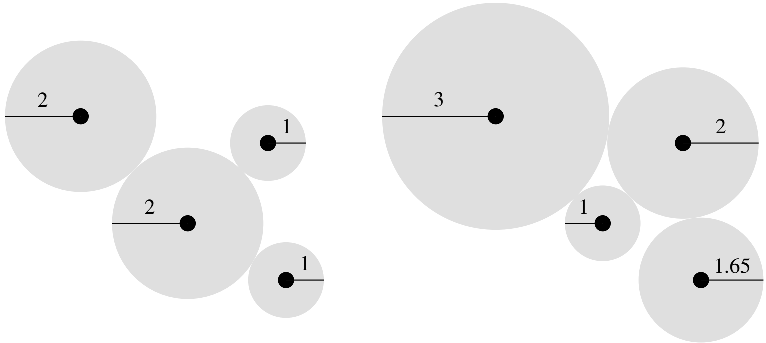


Figure 10.40 Optimal 1-tree for instance in Figure 10.5 with $\lambda(10, 10, 1, 0, 10, 7, 5, 3)$. Resulting lower bound = 81.



(a) Control zones with $2\sum_{j \in N} r_j = 12$. (b) Control zones with $2\sum_{j \in N} r_j = 15.3$.

Figure 10.41 Control zones.

10.6.2 Control Zones

Suppose the nodes in N are located on the plane and c_{ij} is the Euclidean distance between nodes i and j—this is the Euclidean TSP. Imagine drawing a disk centered at each node in N so that the disks do not overlap (Figure 10.41(a)). Any tour must enter and exit disk j on its way to and from node j, for every $j \in N$. This means that the length of any tour must be at least $2\sum_{j \in N} r_j$, where r_j is the radius of disk j. These disks—called *control zones*—therefore provide a lower bound on the length of the optimal TSP tour.

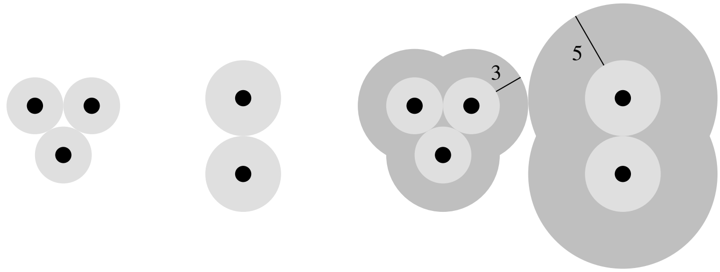
In Figure 10.41(a), we have $2\sum_{j \in N} r_j = 12$. This is not the only possible "packing" of control zones around these nodes, and other packings may produce tighter bounds. For example, the control zones in Figure 10.41(b) have $2\sum_{j \in N} r_j = 15.3$, close to the optimal tour length of 15.7. This gives rise to an optimization problem: We wish to choose radii for the control zones to maximize twice their sum, while ensuring that no two control zones overlap:

$$\text{maximize} \quad 2\sum_{j \in N} r_j \tag{10.37}$$

$$\text{subject to} \quad r_i + r_j \le c_{ij} \qquad \forall i, j \in N \tag{10.38}$$

This optimization problem turns out to be the dual of the LP relaxation of the 2-matching relaxation; that is, if we remove the subtour-elimination constraints from (10.7)–(10.10), allow the x_{ij} to be fractional, and then take the dual, we will get (10.37)–(10.38).

We can't hope for the optimal control-zone bound to be particularly good, given that it is two levels of relaxation away from the original TSP. But we can improve it significantly. Consider the nodes in Figure 10.42(a). The control zones cannot be enlarged further without overlapping. Any feasible tour must pass through the gap in between the two clusters, but the control-zone bound will not account for this distance. However, we can draw bands around the clusters, called *moats*, such that the bands from two different clusters do not overlap. (See Figure 10.42(b).) Any tour must pass through every moat twice, once when approaching the cluster and once when departing it, so we can add twice the sum of the moat widths to the lower bound. This helps a great deal, resulting in lower bounds that are often quite tight (Applegate et al. 2007). In effect, adding moats eliminates subtours,



(a) Gap between control zones. (b) Filling the gap with moats.

Figure 10.42 Moats.

bringing the 2-matching relaxation closer to the original TSP and the control-zone bound closer to the optimal TSP value.

The use of control zones and moats was proposed by Jünger and Pulleyblank (1993). In addition to providing useful lower bounds, control zones also make for beautiful pictures when rendered in color—see especially the Concorde iOS app (Cook 2018b) and the GEODUAL software by Jünger et al. (2009).

10.6.3 Integrality Gap

From Theorem 10.13, we know that

$$z^{CH} \le \frac{3}{2} z^*,$$

where z^{CH} is the length of the tour returned by Christofides' heuristic. It turns out that this is true even if we replace the discrete problem (10.7)–(10.10) with its LP relaxation (Wolsey 1980, Shmoys and Williamson 1990), that is:

$$z^{CH} \le \frac{3}{2} z^{LP}. \tag{10.39}$$

Since $z^* \le z^{CH}$, this gives a fixed worst-case bound on the integrality gap, defined as the ratio between the IP value and the LP relaxation value.

Lemma 10.18 *For any TSP instance satisfying the triangle inequality,*

$$\frac{z^*}{z^{LP}} \le \frac{3}{2}.$$

This bound is not believed to be tight, however. In fact, there are no known instances whose integrality gap exceeds $4/3$. This has given rise to the "$4/3$ conjecture" (e.g., Goemans (1995)):

Conjecture 10.1 *For any TSP instance satisfying the triangle inequality,*

$$\frac{z^*}{z^{LP}} \le \frac{4}{3}.$$

Benoit and Boyd (2008) prove that the $4/3$ conjecture holds for all networks with up to 10 nodes. Moreover, they give a family of instances for which z^*/z^{LP} approaches $4/3$ asymptotically as the number of nodes increases, which means that no *smaller* bound than $4/3$ is possible. Boyd and Elliott-Magwood (2010) extend this result to networks with up to 12 nodes.

Recall that the 2-matching relaxation is the formulation (10.7)–(10.10) without the subtour-elimination constraints (10.9) (Section 10.3.3). Like the LP relaxation, the 2-matching relaxation provides a lower bound on the optimal TSP length. Boyd and Carr (1999) prove that the ratio between z^{2M} (the optimal objective value of the 2-matching relaxation) and z^{LP} is no more than $\frac{4}{3}$. Boyd and Carr (2011) conjecture that this ratio is in fact $\frac{10}{9}$, and Schalekamp et al. (2014) prove their conjecture, i.e., that

$$\frac{z^{2M}}{z^{LP}} \leq \frac{10}{9}. \tag{10.40}$$

10.6.4 Approximation Bounds

From Theorem 10.13, we know that it is possible to find a polynomial-time heuristic with a fixed worst-case bound of 1.5 (i.e., a 1.5-approximation algorithm) for the metric TSP. But it is not known whether a better heuristic, with a better bound, is possible. Since the metric TSP is NP-hard (Papadimitriou 1977), no approximation algorithm can achieve a bound of 1 (unless P = NP), and in fact it is known that no approximation algorithm can reduce the bound to $123/122 \approx 1.008$ (Karpinski et al. 2013) (unless P = NP). Whether the best theoretical bound for an approximation algorithm is $123/122$ or 1.5 or somewhere in between remains an open question.

If the c_{ij} equal Euclidean distances—the Euclidean TSP—then we can get arbitrarily close to 1 in polynomial time. Arora (1998) showed that the Euclidean TSP has a *polynomial-time approximation scheme* (PTAS)—a family of polynomial-time algorithms that, for any $\eta > 1$, can approximate the problem to within $1 + 1/\eta$. For fixed η, the algorithm must be polynomial, though the complexity can be different for different η. Arora's PTAS for the Euclidean TSP runs in $O(n(\log n)^{O(\eta)})$ time. So, for example, to achieve a worst-case bound of $1.008 = 1 + 1/125$, the run time would increase roughly as $n(\log n)^{125}$—still polynomial, but slow. The PTAS is therefore of more theoretical than practical interest.

Interestingly, even though the Euclidean TSP is a special case of the metric TSP, the metric TSP does *not* have a PTAS unless P = NP (Arora et al. 1998).

10.6.5 Tour Length as a Function of n

A common rule of thumb is that the optimal length of a TSP tour through n random points is proportional to $\sqrt{n}$ (as n gets large). Obviously, the constant of proportionality depends on the size of the region in which the points are located. If the points are located in the unit square, then the constant of proportionality is denoted β and called the *TSP constant*. This result was formalized in a famous result by Beardwood et al. (1959):

Theorem 10.19 *Suppose that the node locations in the TSP are uniformly distributed in the unit square. Let z^*_n be the optimal objective function value for the problem with n*

nodes. Then

$$\lim_{n\to\infty} \frac{1}{\sqrt{n}} z_n^* = \beta \tag{10.41}$$

almost surely (a.s.).

Proof. Omitted; see Beardwood et al. (1959) or Karp and Steele (1985). ∎

If the nodes are located in a general region with area A, then (10.41) becomes

$$\lim_{n\to\infty} \frac{1}{\sqrt{n}} z_n^* = \beta\sqrt{A}. \tag{10.42}$$

No one knows the value of β, but Cook (2012) reports on an ongoing experiment in which he and his colleagues have solved over 600 million randomly generated Euclidean TSP instances and used the results to estimate that $\beta \approx 0.712$.

Haimovich and Rinnooy Kan (1985) prove that if the nodes in N are contained in a planar region with area A and perimeter p, then the following upper bound holds on z^*:

$$z^* \le \sqrt{2nA} + \frac{3}{2}p. \tag{10.43}$$

This bound is not particularly tight. If the region is the unit square, for example, then the right-hand side of (10.43) is $\sqrt{2n} + 6$, which is much larger than the estimate $0.712\sqrt{n}$ from Theorem 10.19.

10.7 WORLD RECORDS

In addition to sparking creativity and dedication among researchers, the TSP has also ignited their competitive spirit. A quest to solve larger and larger TSP instances has been ongoing for decades. To count as "solved," an instance must have both a solution and a proof of optimality—that is, both a feasible solution and a lower bound with equal value.

A reasonable start date for this friendly competition is 1954, when Dantzig et al. (1954) solved a 49-node instance consisting of one city from each of the United States (at the time, there were only 48) plus Washington, DC. Their solution has a total length of (coincidentally) 12,345 miles. They solved the problem by first eliminating seven consecutive cities from the east coast (since an optimal tour through these cities is easy) and then using a manual cutting-plane method to solve the remaining 42-node instance. Their 42-node solution is pictured in Figure 10.43(a).

New records were established gradually over the subsequent decades. Held and Karp (1971) solved a 64-node random instance using the method described in Section 10.6.1. Camerini et al. (1975) solved a 67-node random instance. In 1977, Grötschel solved an instance consisting of 120 German cities using branch-and-cut (reprinted as Grötschel (1980b)); see Figure 10.43(c). Lin and Kernighan (1973) solved a 318-node instance derived from a drilling application using their heuristic, discussed in Section 10.5.1; Padberg and Rinaldi (1987) used branch-and-cut to solve an instance consisting of 532 AT&T switches in the United States (Figure 10.43(b)); and in 1987, Grötschel and Holland solved a 666-node instance consisting of locations in the United States (published in Holland 1987, Grötschel and Holland 1991), again using branch-and-cut. Another major breakthrough occurred in 1987, when Padberg and Rinaldi announced the solution of a 2392-node instance arising from printed circuit board drilling (published in Padberg and Rinaldi 1991); see Figure 10.43(d). They accomplished this using emerging supercomputing resources as well as new cutting planes.

The Concorde solver by Applegate et al. (2006) set and shattered a number of records in the late 1990s and early 2000s. The solver's first major milestone was the solution of a 13,509-node United States data set in 1998, using 4.1 years of CPU time in a parallel computing environment. Concorde's developers announced the solution of a 15,112-city German instance (requiring 22.6 CPU years on 110 processors) in 2001 and a 24,978-node Swedish instance (84.8 CPU years) in 2004. Concorde solved an 85,900-node instance based on very-large-scale integration (VLSI) design in 2006 (136 CPU years), and this is the current world record, as of this book's printing. This instance had been *nearly* solved for 15 years: A solution of length 142,514,146 was found in 1991; another of length 142,382,641 (0.09% shorter) was found in 2004; and that solution was proved optimal in 2006.

Several large instances remain unsolved. A 100,000-node instance representing a discretization of the *Mona Lisa* has, at the time of this printing, a known solution of length 5,757,191 and a lower bound of 5,757,084, for a 0.0019% optimality gap. A 115,475-node instance consisting of all US cities has a current best known solution of 6,204,999 miles. And the massive "World TSP" data set consisting of all populated cities or towns in the world (plus a few research bases in Antarctica), has 1,904,711 nodes. The current best solution for this instance, from March 2018, has length 7,515,772,107, and the best known lower bound is 7,512,218,268, for a 0.0473% optimality gap.

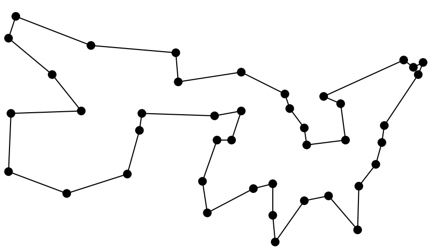
See Cook (2012, 2018a) for much more about TSP world records.

CASE STUDY 10.1 Routing Meals on Wheels Deliveries

The Meals on Wheels (MOW) program delivers hot lunches to the elderly and others who are unable to shop or cook for themselves. In the early 1980s, MOW (or, more precisely, Senior Citizens, Inc., a nonprofit organization in Atlanta that delivers MOW), worked with researchers at Georgia Tech to implement a heuristic for meal-delivery routes for its drivers. The project was described by Bartholdi et al. (1983), whose discussion we follow here.

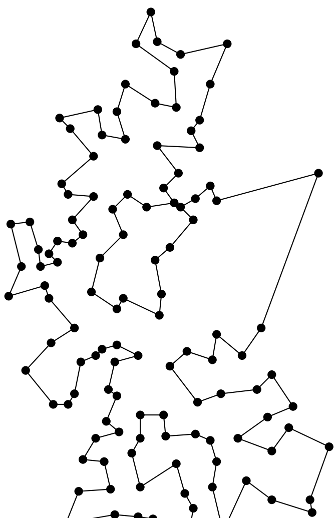
MOW's client list changes often—roughly 14% each month—as clients fall ill or recover, move to assisted living facilities or a family member's home, and so on. Moreover, MOW's funding is highly variable, and the number of clients that the organization can serve may increase or decrease sharply with funding levels. This makes planning meal deliveries challenging, especially since, like most charitable organizations, MOW has no in-house operations research expertise.

This makes the *space-filling curve heuristic* by Bartholdi and Platzman (1982) particularly appealing for MOW. The space-filling curve heuristic can be executed easily and does not even require a computer. It can easily accommodate changes in the list of nodes to visit, and it is reasonably accurate, with solutions that are about 25% longer than optimal, on average. A *space-filling curve* is a curve that is built recursively in a fixed region, becoming longer and more intricate at each iteration; in the limit, the curve fills the entire area of the region. Figure 10.44(a) shows the so-called Hilbert space-filling curve after four iterations.

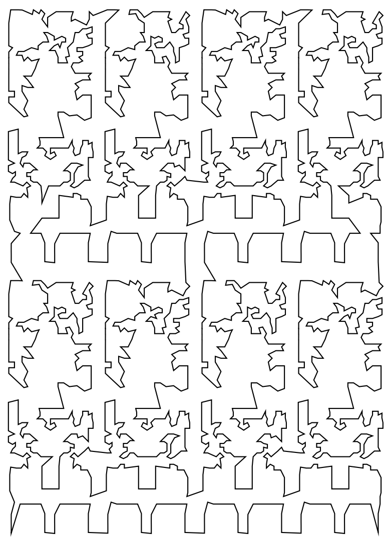


(a) Solution to 42-node reduction of 49-node instance by Dantzig et al. (1954). Adapted from Dantzig.

(b) Solution to 532-node AT&T instance by Padberg and Rinaldi (1987). Reproduced with permission of Elsevier.

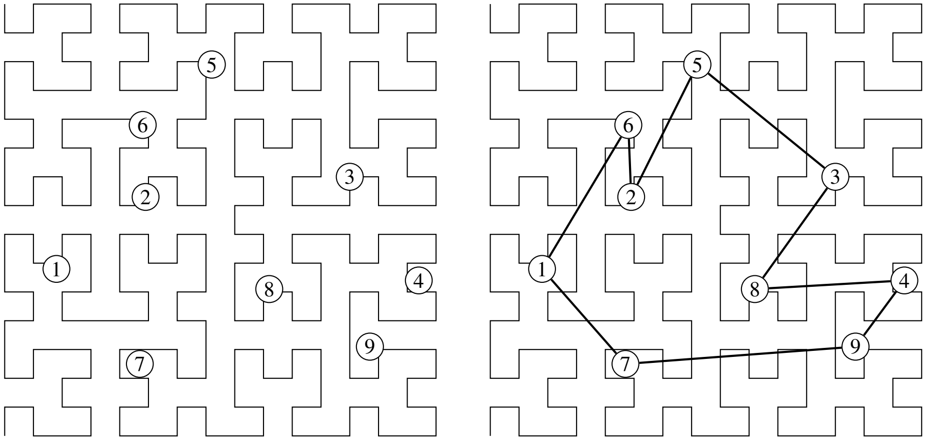


(c) Solution to 120-node German instance by Grötschel (1980b). Reproduced with permission of Springer.



(d) Solution to 2392-node circuit board instance by Padberg and Rinaldi (1991). Copyright ©1991 Society for Industrial and Applied Mathematics. Reprinted with permission. All rights reserved.

Figure 10.43 TSP world records.



(a) Customer nodes along space-filling curve. (b) Solution returned by space-filling curve heuristic.

Figure 10.44 Space-filling curve heuristic.

The space-filling curve heuristic first constructs (a few iterations of) a space-filling curve in the region containing the customer nodes. Each customer node is considered to lie on the curve, at the point that it is closest to. The nodes are then sequenced according to their position along the space-filling curve. For example, in Figure 10.44(a), start following the space-filling curve from the top-left corner. As you travel (with your eye or finger) along the space-filling curve, the customer nodes are encountered in the following sequence: $6 \to 2 \to 5 \to 3 \to 8 \to 4 \to 9 \to 7 \to 1$. The heuristic therefore returns a TSP route consisting of that sequence of nodes; see Figure 10.44(b). (This solution has a total length of 44.95, which is 22.6% longer than the optimal tour.)

The Georgia Tech team implemented the space-filling curve heuristic for MOW as follows. First, they plotted the (x, y) coordinates of each of the clients' homes on the space-filling curve. They assigned a unique value θ to each client, representing the relative position of the client on the space-filling curve. For example, a client located a quarter of the way from the beginning of the curve to the end has $\theta = 0.25$. The clients were then sorted in order of θ. As clients are added or removed, they can simply be added to or removed from the sorted list, which MOW maintained easily using a Rolodex. All of the steps so far—mapping the clients, assigning them θ values, and sorting them—can be done ahead of time. On each delivery day, all that remains is to divide the sorted list of clients into (typically) four batches, one for each delivery driver. The drivers can then simply follow the route order given in the list. The researchers estimated that the new heuristic saved approximately 13% in total travel distance.

PROBLEMS

10.1 (TSP Construction Heuristics #1) Use each of the construction heuristics listed below to find solutions for the network shown in Figure 10.45. Begin the insertion heuristics at node 1.

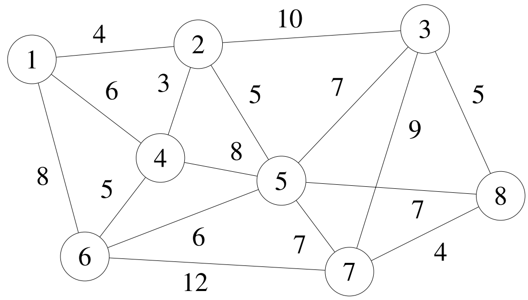


Figure 10.45 TSP instance for Problems 10.1 and 10.2. Edges that are not pictured have lengths given by shortest-path distances.

In the figure, edges that are not pictured have lengths given by shortest-path distances. The optimal tour on this network is $1 \to 2 \to 3 \to 8 \to 7 \to 5 \to 6 \to 4$, with total length 47. For each heuristic, report the tour found and its length. For heuristics for which worst-case error bounds are available, confirm that the ratio of the length of the tour returned by the heuristic and the length of the optimal tour is no greater than the bound.

a) Nearest neighbor
b) Nearest insertion
c) Cheapest insertion
d) Farthest insertion
e) Minimum spanning tree
f) Christofides

10.2 (TSP Improvement Heuristics #1) Consider the following tour on the network given in Figure 10.45: $1 \to 6 \to 4 \to 2 \to 3 \to 7 \to 5 \to 8$. Perform two iterations of each of the improvement heuristics listed below (starting from the original tour for each heuristic). Each iteration consists of one move that improves the solution, but it need not be the optimal such move. Indicate the nodes and/or edges involved in each move, as well as the tour that results and its cost.

a) 2-opt
b) 3-opt
c) Or-opt

10.3 (TSP Construction Heuristics #2) Repeat Problem 10.1 using the network given in Figure 10.46. Edges that are not pictured have lengths given by shortest-path distances. The optimal tour on this network is $1 \to 2 \to 3 \to 4 \to 6 \to 7 \to 5$, with total length 395.

10.4 (TSP Improvement Heuristics #2) Repeat Problem 10.2 to improve the following tour for the network given in Figure 10.46: $1 \to 5 \to 4 \to 6 \to 7 \to 2 \to 3$.

10.5 (TSP Construction Heuristics #3) Repeat Problem 10.1 using the network given in Figure 10.47. Coordinates for the nodes in the figure are given in Table 10.1. Distances between nodes are Euclidean. The optimal tour on this network is $1 \to 4 \to 9 \to 10 \to 2 \to 5 \to 8 \to 3 \to 7 \to 6$, with total length 33.45.

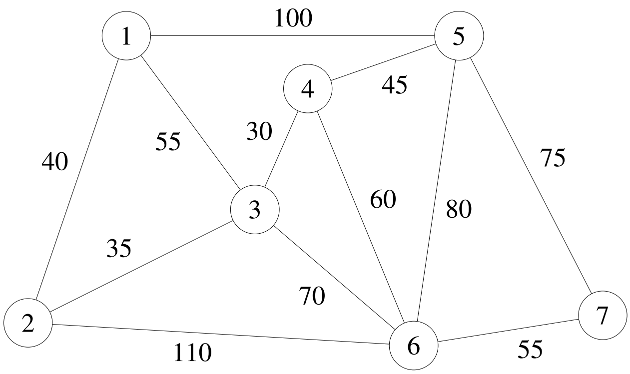


Figure 10.46 TSP instance for Problems 10.3 and 10.4. Edges that are not pictured have lengths given by shortest-path distances.

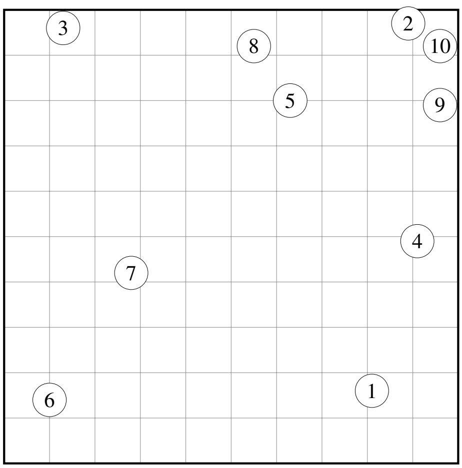


Figure 10.47 TSP instance for Problems 10.5 and 10.6. Distances are Euclidean.

Table 10.1 Node coordinates for Problems 10.5 and 10.6.

i	x_i	y_i	i	x_i	y_i
1	8.1	1.6	6	1.0	1.4
2	8.9	9.7	7	2.8	4.2
3	1.3	9.6	8	5.5	9.2
4	9.1	4.9	9	9.6	7.9
5	6.3	8.0	10	9.6	9.2

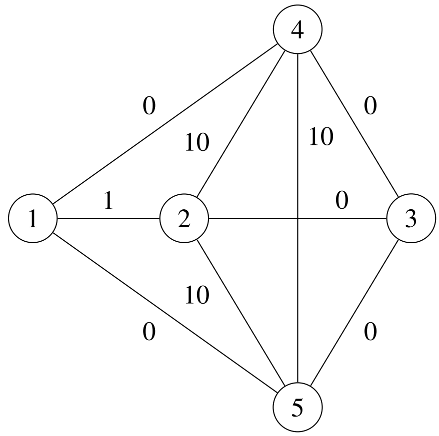


Figure 10.48 TSP instance for Problem 10.16. Edges that are not pictured have a length of ∞.

10.6 (TSP Improvement Heuristics #3) Repeat Problem 10.2 to improve the following tour for the network given in Figure 10.47: $1 \to 4 \to 7 \to 6 \to 3 \to 9 \to 5 \to 2 \to 10 \to 8$. Coordinates for the nodes in the figure are given in Table 10.1. Distances between nodes are Euclidean.

10.7 (GENI Insertion Can Shorten Tour) Construct a small example that demonstrates that a GENI insertion can decrease the tour length.

10.8 (Proof of Handshaking Lemma) Prove Lemma 10.12.

10.9 (Proof that Euclidean Tours Do Not Cross Themselves) Prove Lemma 10.14.

10.10 (Proof of Comb-Inequality Theorem) Prove Theorem 10.4.

10.11 (Proof of Alternate Comb Inequality) Prove that the alternate 2-matching inequality (10.16) also holds for combs as defined in Theorem 10.4.

10.12 (MST Heuristic Bound is Tight) Prove that the bound of 2 for the MST heuristic in Theorem 10.11 is tight.

10.13 (Christofides' Heuristic Bound is Tight) Prove that the bound of $3/2$ for Christofides' heuristic in Theorem 10.13 is tight.

10.14 (Proof that P = NP?) What is the logical flaw in the following argument?

> For any distance matrix c that does not satisfy the triangle inequality, we can always add a sufficiently large constant M to every element in the matrix so that the new matrix, c', *does* satisfy the triangle inequality. Moreover, a tour T_1 is shorter than T_2 under c if and only if it is shorter under c'. Apply Christofides' heuristic on the revised instance c'; then the resulting solution is no more than $\frac{3}{2}$ worse than optimal, by Theorem 10.13. Since this works for any TSP instance, even those that do not satisfy the triangle inequality, by Theorem 10.5, P = NP.

10.15 (Proof of Lemma 10.16) Prove Lemma 10.16.

10.16 (Held–Karp Bound #1) Consider the network in Figure 10.48. Edges that are not pictured have a length of ∞.

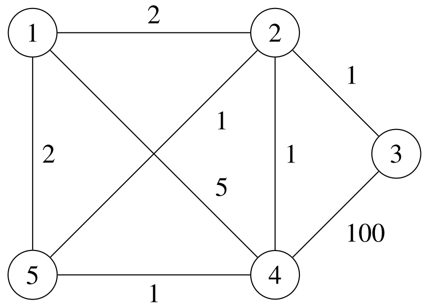


Figure 10.49 TSP instance for Problem 10.17. Edges that are not pictured have a length of ∞.

a) What is the optimal TSP tour on this network, and what is its total length, z^*? What is the optimal 1-tree $\hat{T}^*$ on this network ("rooted" at node 1), and what is its total length, $z(\hat{T}^*)$?

b) Find the best Held–Karp lower bound you can. That is, find weights to add to one or more nodes so that z^* and the right-hand side of (10.33) are as close as possible.

10.17 (Held–Karp Bound #2) Repeat Problem 10.16 for the network in Figure 10.49.

10.18 (TSP with Required Edges) Consider a version of the TSP in which we are given a set M of edges that *must* be part of the tour. Assume that M is a matching.

a) Modify Christofides' heuristic to solve this problem.

b) Show that the performance of your heuristic from part (a) has a fixed worst-case bound of $3/2$.

10.19 (Equivalence of Subtour-Elimination Constraints) Prove that (10.5) and (10.6) are equivalent.

10.20 (Miller–Tucker–Zemlin Subtour-Elimination Constraints) Miller et al. (1960) propose introducing new decision variables u_i ($i = 2, \dots, n$) for the TSP and then replacing the subtour-elimination constraints (10.9) with

$$u_i - u_j + (n-1)x_{ij} \le n-2 \quad \forall i, j = 2, \dots, n, i \neq j \tag{10.44}$$

$$1 \le u_i \le n-1 \quad \forall i = 2, \dots, n. \tag{10.45}$$

Prove that the resulting formulation is valid, i.e., that (10.44)–(10.45) prohibit subtours. (These constraints are called the Miller–Tucker–Zemlin, or MTZ, subtour-elimination constraints.)

10.21 (Stengthened MTZ Constraints) Prove that (10.44) can be replaced with the tighter constraint

$$u_i - u_j + (n-1)x_{ij} + (n-3)x_{ji} \le n-2 \quad \forall i, j = 2, \dots, n, i \neq j \tag{10.46}$$

in the MTZ subtour-elimination constraints (see Problem 10.20).

10.22 (Sequence of Convex Hull Nodes) Let $N' \subseteq N$ be the set of nodes that lie on the convex hull of N in an instance of the Euclidean TSP. Prove that every optimal tour visits the nodes in N' in the same sequence in which they occur on the convex hull.

10.23 (Estimating β) Perform a computational experiment to estimate the TSP constant β from Theorem 10.19 by generating random Euclidean TSP instances on the unit square, solving them optimally, and calculating the ratio between the optimal tour length and $\sqrt{n}$, where n is the number of nodes.

10.24 (Nearest Neighbor for Asymmetric TSP) Prove that, for the asymmetric TSP (in which the distance matrix need not be symmetric) in which the triangle inequality holds, the nearest neighbor heuristic can produce solutions for which

$$\frac{z^{NN}}{z^*} > \frac{n}{2},$$

where n is the number of nodes.

10.25 (TSP with Pickup and Delivery) Consider the following variant of the TSP. We must visit a set S of source nodes and a set D of destination nodes, with $|S| = |D|$, transporting a single type of item from the source nodes to the destination nodes. Any source node can supply any destination node. Each source unit has 1 unit of supply, each destination node demands 1 unit, and the vehicle has a capacity of 1 unit. Adapt the cheapest insertion heuristic to solve this problem. Explain your heuristic in words as well as in pseudocode.

10.26 (Prize-Collecting TSP) In the *prize-collecting TSP*, there is no constraint requiring every node to be on the tour, but there is a reward π_i for visiting node i. The objective is to minimize the total tour length minus the total rewards for nodes visited.

a) Formulate the prize-collecting TSP as a linear integer programming problem.

b) Propose a construction heuristic for the prize-collecting TSP. Explain your heuristic in words as well as in pseudocode.

10.27 (1-Tree Formulation is a Relaxation) The problem of finding an optimal 1-tree can be formulated as follows:

$$\text{minimize} \quad \sum_{i,j \in N} c_{ij} x_{ij} \tag{10.47}$$

$$\text{subject to} \quad \sum_{i,j \in N} x_{ij} = n \tag{10.48}$$

$$\sum_{i \in N} x_{1i} = 2 \tag{10.49}$$

$$\sum_{\substack{i \in S, j \in \bar{S} \setminus \{1\} \text{ or} \\ i \in \bar{S} \setminus \{1\}, j \in S}} x_{ij} \geq 1 \qquad \forall S \subseteq N \setminus \{1\} : 1 \leq |S| \leq n-1 \tag{10.50}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i \in N, \forall j \in N \tag{10.51}$$

Prove that this formulation is a relaxation of the TSP formulation (10.7)–(10.10).

10.28 (Subtour-Elimination Constraints for $|S| = 2$) Prove that it is sufficient to replace the "for all" part of constraints (10.9) with $\forall S \subseteq N : 3 \leq |S| \leq n-3$.

10.29 (Pseudocode for GENI Insertion) Write pseudocode for the procedure to find the best Type I GENI insertion; that is, to implement step 5 in Algorithm 10.4 for Type I insertions.

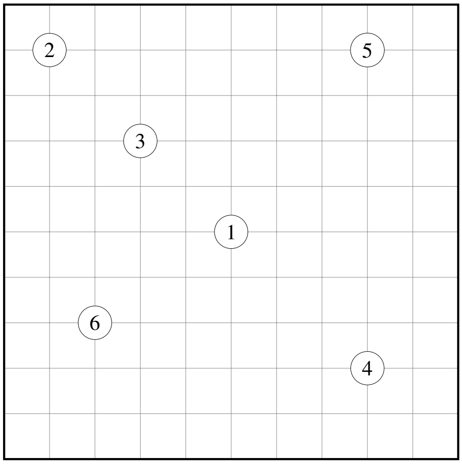


Figure 10.50 TSP instance for Problem 10.32. Distances are Euclidean.

10.30 (Pseudocode for Greedy Heuristic) The *greedy heuristic* for the TSP works as follows: Initialize the tour with the edge (i, j) that minimizes c_{ij}. At each subsequent iteration, add to the tour the shortest edge that is not on the tour, does not create a subtour, and does not create a degree-3 node. (See, e.g., Ong and Moore (1984).) Write pseudocode for this heuristic.

10.31 (Removing Cities from TSP) Suppose we solve the TSP on an arbitrary instance (which satisfies the triangle inequality but is not necessarily Euclidean) and get the optimal tour. Now we decide we do not need to visit one of the cities. Prove or disprove the following claim: The sequence of nodes on the original optimal tour remains optimal after we remove a city; that is, we can simply "close up" the tour around the city we removed.

10.32 (Optimal Control Zones) Find the optimal control zones for the TSP instance in Figure 10.50. The nodes are located at integer coordinates, and the distances among them are Euclidean. Report the radii of each of the control zones, and their sum.

CHAPTER 11

THE VEHICLE ROUTING PROBLEM

11.1 INTRODUCTION TO THE VRP

11.1.1 Overview

The *vehicle routing problem* (VRP) is concerned with optimizing a set of routes, all beginning and ending at a given node (called the *depot*), to serve a given set of *customers*. The VRP was first introduced by Dantzig and Ramser (1959). It is a multi-vehicle version of the traveling salesman problem (TSP), and is therefore more applicable in practice since most organizations with substantial delivery operations use multiple vehicles simultaneously. Of course, it is also more difficult than the TSP since it involves decisions about how to assign customers to routes, in addition to how to optimize the sequence of nodes on each route. As a result, today's "hard" VRP instances tend to involve, say, hundreds of nodes, whereas hard instances of the TSP involve thousands or tens of thousands of nodes.

Figure 11.1 shows the optimal solution to a VRP instance called `eil51` from the TSPLIB data set repository (Reinelt 1991) and originally from Christofides and Eilon (1969). The depot is near the center of the region, marked by a square, while the customers are drawn as circles. Each node has a coordinate in $\mathbb{R}^2$, and the distances between pairs of nodes are Euclidean. The demands range from 3 to 41, and the vehicle capacity is 160. Note that the optimal VRP solution involves routes that cross each other. Of course, just as in the TSP, it is never optimal for an individual route to cross *itself* if the distances are Euclidean, since each individual route in a VRP solution is a TSP tour on the nodes in the route.

Fundamentals of Supply Chain Theory, Second Edition. Lawrence V. Snyder and Zuo-Jun Max Shen.

Companion website: www.wiley.com/go/Snyder/SupplyChainTheory

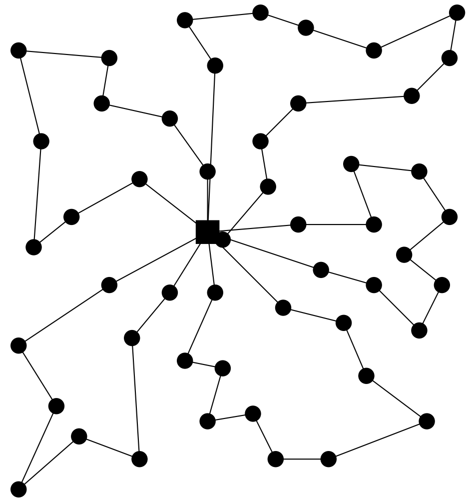


Figure 11.1 Optimal solution to `eil51` VRP instance. Total distance = 521.

The VRP is used to model *less-than-truckload* (LTL) deliveries, in which a single vehicle delivers goods to multiple customer nodes before returning to the depot. In contrast, the facility location models in Chapter 8 assume *truckload* (TL) deliveries, in which a vehicle delivers to only a single node. In addition to differences in the shape of the delivery routes, TL and LTL shipments have different cost structures for shippers. Typically, when a firm ships products TL, it pays a fixed cost for the shipment; the fixed cost depends on the origin and destination points, but not on the quantity of goods being shipped. In contrast, LTL shipments are charged based on weight or volume, as well as on origin and destination. Companies such as FedEx and UPS are LTL carriers.

Not surprisingly, the VRP is an immensely important problem for such carriers. For example, UPS's route-optimization system, called On-Road Integrated Optimization and Navigation (ORION), saves the company an estimated \$300–\$400 million per year and is sometimes described as the world's largest operations research project (Holland et al. 2017). UPS and ORION won the prestigious INFORMS Edelman Award in 2016. (See Case Study 11.1.)

We will formulate the VRP in Section 11.1.3, and then discuss exact and heuristic algorithms in Sections 11.2 and 11.3, respectively. For a more thorough discussion of the VRP, see the reviews by Laporte and Nobert (1987), Laporte (1992b), Laporte et al. (2000) and Cordeau et al. (2002), among others, and the books by Toth and Vigo (2001a) and Golden et al. (2008). Several web sites compile data sets and other useful information about the VRP, including the VRP Repository[1] and the NEO Research Group at the University of Malaga.[2]

[1] `http://www.vrp-rep.org/`
[2] `http://neo.lcc.uma.es/vrp/location/`

11.1.2 Notation and Assumptions

We are given a set $N = \{0, \ldots, n\}$ of nodes and a symmetric distance matrix c_{ij} that satisfies the triangle inequality. Node 0 is the depot. Let $N^- = N \setminus \{0\}$ be the set of customer nodes. Each node $i \in N^-$ has a demand $d_i \geq 0$, with $d_0 = 0$. For $S \subseteq N^-$, let

$$d(S) = \sum_{i \in S} d_i.$$

Some of the models and algorithms we consider below will assume that there are exactly K vehicles available at the depot, and that all of them must be used, while others allow the number of vehicles to be unrestricted. In either case, we assume that the vehicles are all identical, each with a capacity of C. We assume that $d_i \leq C$ for all $i \in N^-$; otherwise, the problem is infeasible.

Other capacity-type constraints are sometimes used instead of, or in addition to, vehicle capacities. For example, in some models, the total distance or time that a vehicle travels is constrained. (If there were no capacity-type constraints and the distances satisfy the triangle inequality, then the VRP would be equivalent to the TSP—why?) The problem we consider is sometimes called the *capacitated vehicle routing problem* (CVRP) to distinguish it from models with these other types of constraints, but we will refer to the problem simply as the VRP.

Many variations of this basic setup are possible. For example, the vehicles may be nonidentical in terms of their capacities or other constraints, or we may incur a fixed cost for each vehicle used. There are more complex extensions, as well, such as:

- *Time windows* during which vehicles must arrive at each customer.
- *Multiple depots* that nodes may be served from, with the assignment of nodes to depots a decision variable.
- *Backhauls*, in which some customers require product to be delivered, others require product to be picked up and brought back to the depot, and delivery customers must come before backhaul customers on a given route.
- *Pickups and deliveries*, in which customers require their shipment to be picked up at one location and delivered to another, both by the same vehicle.
- *Periodic* models in which a given customer must be visited a fixed number of times per week (or month, etc.).

We will discuss some of these variants and extensions in Section 11.5.

11.1.3 Formulation of the VRP

As in the TSP, we will define a decision variable x_{ij} that equals 1 if a route goes from i to j *or* from j to i, for $i, j \in N$. For the depot, we allow $x_{0j} \in \{0, 1, 2\}$, where $x_{0j} = 2$ indicates a single-customer route. (If single-customer routes are prohibited, we can simply require $x_{0j} \in \{0, 1\}$.) The decision variables are defined only for $i < j$, but, as with the TSP, we will often omit the requirement $i < j$ when writing summation and constraint indices.

In this section, we assume that exactly K vehicles must be used to serve the nodes in N^-. For a subset $S \subseteq N^-$, let $v(S) \geq 1$ be the minimum number of vehicles required to serve all nodes in S. We discuss the calculation of $v(S)$ below.

The VRP can be formulated as an integer programming (IP) model as follows:

$$\text{(VRP)} \quad \text{minimize} \quad \sum_{i,j \in N} c_{ij} x_{ij} \tag{11.1}$$

$$\text{subject to} \quad \sum_{i \in N} x_{ih} + \sum_{j \in N} x_{hj} = 2 \qquad \forall h \in N^- \tag{11.2}$$

$$\sum_{j \in N^-} x_{0j} = 2K \tag{11.3}$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - v(S) \qquad \forall S \subseteq N^- : S \neq \emptyset \tag{11.4}$$

$$x_{ij} \in \{0,1\} \qquad \forall i,j \in N^- \tag{11.5}$$

$$x_{0j} \in \{0,1,2\} \qquad \forall j \in N^- \tag{11.6}$$

(This is a slightly less general version of the formulation presented by Laporte et al. (1985).) The objective function (11.1) calculates the total length of the routes. Constraints (11.2) require exactly two edges to be incident to each node except the depot, and constraint (11.3) requires $2K$ edges to be incident to the depot. Constraints (11.4) are called *capacity-cut constraints* (or sometimes just *capacity constraints*); they are a generalization of subtour-elimination constraints. Equivalently, we could use either

$$\sum_{\substack{i \in S, j \in \bar{S} \text{ or} \\ i \in \bar{S}, j \in S}} x_{ij} \geq 2v(S) \qquad \forall S \subseteq N^- : S \neq \emptyset \tag{11.7}$$

or

$$\sum_{i \in S, j \in \bar{S}} x_{ij} \geq v(S) \qquad \forall S \subseteq N^- : S \neq \emptyset. \tag{11.8}$$

Constraints (11.5) and (11.6) enforce the integrality and bounds on the x_{ij}. This formulation is known as a *two-index formulation* since the decision variables each have two indices.

This formulation is remarkably similar to the TSP formulation (10.7)–(10.10), despite the fact that the VRP allows multiple routes (and must ensure that each is connected to the depot) and has capacity restrictions. The capacity-cut constraints (11.4) do quite a bit of heavy lifting, ensuring both the depot-connectedness *and* the capacity-feasibility of every route. To see this, suppose first that there is a route that is not connected to the depot—in other words, a subtour. Let S be the nodes on this route. Then the number of edges contained within S (the left-hand side of (11.4)) equals $|S|$, which violates (11.4) since $v(S) \geq 1$. Now suppose that there is a route that violates the capacity constraint. Again, let S be the nodes on this route, excluding the depot. Then $\sum_{i,j \in S} x_{ij} = |S| - 1$, but since the route is over capacity, $v(S) \geq 2$ and (11.4) is violated.

The question remains how to calculate $v(S)$. An exact calculation of $v(S)$ requires solving the *bin-packing problem*. In the bin-packing problem, we are given a set of objects, each with a given weight (or other measure of size). The objective is to "pack" the objects into bins, each of which has a fixed capacity, minimizing the total number of bins used. Therefore, $v(S)$ is equal to the optimal objective value of the bin-packing problem for a set

S of objects of weight d_i and bins of capacity C. Unfortunately, the bin-packing problem is NP-hard, so it is common to replace $v(S)$ with a lower bound; one easy bound is

$$\left\lceil \frac{d(S)}{C} \right\rceil. \tag{11.9}$$

In fact, replacing $v(S)$ with the lower bound (11.9) in the capacity-cut constraints (11.4) or (11.7) retains the validity of the IP formulation (Cornuejols and Harche 1993), though it weakens its LP relaxation. Obviously, whether we use (11.9) or the bin-packing problem, it is impractical to calculate $v(S)$ for all $S \subseteq N^-$ since there are exponentially many such sets. Then again, it is impractical to enumerate all of the capacity-cut constraints (11.4) anyway; so $v(S)$ can be calculated as needed when a given constraint is added.

The two-index formulation (11.1)–(11.6) is simple but fairly inflexible. Although one could look at a solution and assign trucks to routes or determine the sequence of nodes on the route, the model itself is not "aware" of these attributes endogenously. Because of this, it cannot be modified to handle time windows, precedence constraints, nonidentical vehicle capacities or capabilities, sequence-dependent costs, or a range of other realistic problem features.

To correct this, we can use a *three-index formulation* that explicitly keeps track of which vehicle is assigned to each route. In particular, $x_{ijk} = 1$ if vehicle k travels directly from i to j, and 0 otherwise. Note that, unlike in the two-index formulation, the x variables indicate which node (i or j) comes before the other. Therefore, x_{ijk} is defined for all $i, j \in N$, not just for $i < j$. We also have a binary variable y_{ik} that equals 1 if vehicle k serves node i and 0 otherwise, for $i \in N^-$. This gives us the following formulation, which is based on the formulation by Fisher and Jaikumar (1981):

$$\begin{aligned}
\text{minimize} \quad & \sum_{k=1}^{K} \sum_{i,j \in N} c_{ij} x_{ijk} && && (11.10)\\
\text{subject to} \quad & \sum_{k=1}^{K} y_{ik} = 1 && \forall i \in N^- && (11.11)\\
& \sum_{k=1}^{K} y_{0k} = K && && (11.12)\\
\sum_{i \in N} x_{ihk} = & \sum_{j \in N} x_{hjk} = y_{hk} && \forall h \in N, \forall k = 1, \ldots, K && (11.13)\\
& \sum_{i \in N} d_i y_{ik} \le C && \forall k = 1, \ldots, K && (11.14)\\
& \sum_{i,j \in S} x_{ijk} \le |S| - 1 && \forall k = 1, \ldots, K, \forall S \subseteq N^- : |S| \ge 2 && (11.15)\\
& x_{ijk} \in \{0, 1\} && \forall i, j \in N^-, \forall k = 1, \ldots, K && (11.16)\\
& y_{ik} \in \{0, 1\} && \forall i \in N^-, \forall k = 1, \ldots, K && (11.17)
\end{aligned}$$

The objective function (11.10) calculates the total route length. Constraints (11.11) require every nondepot node to be served by exactly one route. Constraint (11.12) requires the depot to be contained on K routes. (This constraint can be removed if K is an upper bound on the number of vehicles but not all K vehicles need to be used.) Constraints (11.13) require y_{hk} to equal 1 if and only if vehicle k traverses exactly one arc into h and one arc out of h. Constraints (11.14) ensure the vehicle capacity is not exceeded. Constraints (11.15) are subtour-elimination constraints, and (11.16)–(11.17) are integrality constraints.

This formulation has an explicit capacity constraint (11.14). In the two-index formulation, in contrast, we cannot calculate the total load on a given vehicle endogenously, so capacity constraints must be imposed via the $v(S)$ parameter in the capacity-cut constraints (11.4). The three-index formulation can easily handle vehicle-dependent capacities: We simply change the right-hand side of (11.14) to C_k, where C_k is the capacity of vehicle k.

The three-index formulation is more difficult to solve than the two-index formulation since there are many more binary variables ($O(n^2K)$ vs. $O(n^2)$), but the added complexity is often compensated for by the increased flexibility. In addition, a three-index model can be converted to a two-index model by setting $x_{ij} = \sum_{k=1}^{K} x_{ijk}$; therefore, any valid inequalities developed for the two-index formulation are also valid for the three-index formulation.

11.2 EXACT ALGORITHMS FOR THE VRP

11.2.1 Dynamic Programming

Eilon et al. (1971) propose a dynamic programming (DP) algorithm for the VRP. Whereas the DP for the TSP acts recursively on the nodes, the VRP algorithm acts recursively on the routes, expressing the optimal distance for a solution that uses k routes in terms of the distance using $k-1$ routes.

For a subset $S \subseteq N^-$, let $c(S)$ be the length of the optimal TSP tour through the depot and the nodes in S if $\sum_{i \in S} d_i \le C$ and ∞ otherwise. Define $\theta(S,k)$ as the minimum possible total distance to deliver to the nodes in S using k routes, or ∞ if the nodes in S cannot be feasibly served by k routes. If $k = 1$, then $\theta(S,k) = c(S)$. Suppose $k > 1$. If we know that one of the k routes serves a customer set $S' \subset S$, then the optimal distance is given by $\theta(S \setminus S', k-1) + c(S')$, where the second term computes the length of the route through S' and the first computes the lengths of the remaining $k-1$ routes. Therefore, we can calculate $\theta(S,k)$ recursively:

$$\theta(S,k) = \begin{cases} c(S), & \text{if } k = 1, \\ \min_{S' \subset S} \{\theta(S \setminus S', k-1) + c(S')\}, & \text{otherwise.} \end{cases} \tag{11.18}$$

If the number of vehicles is fixed to K, then the total length of the optimal VRP solution is given by $\theta(N^-, K)$. If the number of vehicles is unrestricted, we can choose the k that minimizes $\theta(N^-, k)$.

Of course, this is not a computationally efficient way to solve the VRP. Not only do we need to enumerate all subsets of N^-, and all subsets of those sets, and so on, but we must also solve the TSP for each of those subsets. One alternative is to "relax" the state space in such a way that the resulting recursion provides a lower bound on the optimal VRP objective function value. For example, we can define the DP recursion in terms of the load represented by the customers in S rather than in terms of the customers themselves. Let $\hat{\theta}(d,k)$ be the optimal distance to use k vehicles to deliver to a set of nodes whose total demand equals d, or ∞ if $d > kC$, i.e., if k vehicles are not sufficient to serve a total demand of d. Let $\hat{c}(d)$ be the length of the optimal TSP tour through the depot and a set of nodes with total demand d, or ∞ if $d > C$. (Actually, calculating an exact value for $\hat{c}(d)$ is itself difficult, so in practice one can replace $\hat{c}(d)$ by a lower bound on it; see Christofides et al. (1981).) Note that for any S,

$$\hat{c}(d(S)) \le c(S) \tag{11.19}$$

since the optimal TSP tour through $S \cup \{0\}$, which has length $c(S)$, is a feasible solution for the problem of finding the optimal tour through the depot and a set of nodes with total demand $d(S)$.

A recursion for $\hat{\theta}(d, k)$ is given by

$$\hat{\theta}(d,k) = \begin{cases} \hat{c}(d), & \text{if } k = 1, \\ \min\limits_{0 \le d' \le d} \left\{ \hat{\theta}(d - d', k-1) + \hat{c}(d') \right\}, & \text{otherwise.} \end{cases} \tag{11.20}$$

The significance of this relaxation is given by the following proposition.

Proposition 11.1 *For any $S \subseteq N^-$ and $k = 1, \dots, K$ (where K is possibly infinite),*

$$\hat{\theta}(d(S), k) \le \theta(S, k). \tag{11.21}$$

In particular, (11.21) *holds for $S = N^-$ and $k = K$, which implies that the relaxed recursion* (11.20) *provides a lower bound on the optimal objective function value of the VRP.*

Proof. By induction on k. Let $S \subseteq N^-$. First suppose $k = 1$. By (11.19),

$$\hat{\theta}(d(S), 1) = \hat{c}(d(S)) \le c(S) = \theta(S, 1).$$

Now suppose (11.21) holds for $k > 1$; we will show it holds for $k + 1$. By (11.18),

$$\begin{aligned} \theta(S, k+1) &= \min_{S' \subset S} \{\theta(S \setminus S', k) + c(S')\} \\ &\ge \min_{S' \subset S} \left\{ \hat{\theta}(d(S \setminus S'), k) + \hat{c}(d(S')) \right\} \end{aligned}$$

(by the induction hypothesis and (11.19))

$$\begin{aligned} &= \min_{S' \subset S} \left\{ \hat{\theta}(d(S) - d(S'), k) + \hat{c}(d(S')) \right\} \\ &\ge \min_{0 \le d' < d(S)} \left\{ \hat{\theta}(d(S) - d', k) + \hat{c}(d') \right\} \end{aligned}$$

(since $d(S')$ is feasible for the minimization over $0 \le d' < d(S)$)

$$= \hat{\theta}(d(S), k+1),$$

as desired. ■

This technique is known as *state-space relaxation*. It was introduced by Christofides et al. (1981), who also introduce several other recursions and relaxations that provide tighter bounds than (11.20). This lower-bounding procedure can be embedded into a branch-and-bound algorithm.

11.2.2 Branch-and-Bound

Because the VRP formulations in Section 11.1.3 have an exponential number of constraints, most branch-and-bound algorithms relax the capacity-cut constraints and solve the resulting problem to obtain lower bounds on the optimal objective function value. For example, suppose we relax the capacity-cut constraints (11.4) in the two-index VRP formulation to obtain the following formulation:

$$\begin{aligned}
\text{minimize} \quad & \sum_{i,j \in N} c_{ij} x_{ij} && && (11.22)\\
\text{subject to} \quad & \sum_{i \in N} x_{ih} + \sum_{j \in N} x_{hj} = b_h && \forall h \in N && (11.23)\\
& x_{ij} \in \{0,1\} && \forall i,j \in N^- && (11.24)\\
& x_{0j} \in \{0,1,2\} && \forall j \in N^- && (11.25)
\end{aligned}$$

Constraints (11.23) combine constraints (11.2) and (11.3) by defining

$$b_h = \begin{cases} 2, & \text{if } i \in N^- \\ 2K, & \text{if } i = 0. \end{cases}$$

The model formulated in (11.22)–(11.25) chooses the minimum-cost set of edges such that every node h has degree b_h. This is known as the *b-matching problem* and is a generalization of the 2-matching problem; it can be solved efficiently (Miller and Pekny 1995). The b-matching problem only provides a lower bound since its solutions may be infeasible for the VRP due to capacity violations or routes that are disconnected from the depot. Miller (1995) proposes a branch-and-bound algorithm based on this b-matching relaxation.

Another relaxation, due to Fisher (1994a), extends the notion of 1-trees (Section 10.6.1) to the VRP. He defines a K-tree as a minimum-cost set of $n + K$ edges that contains every node, and he further focuses on K-trees in which the depot has degree $2K$. In every VRP solution there are $n + K$ edges and the depot has degree $2K$, and so the problem of finding an optimal degree-constrained K-tree is a relaxation of the VRP. This degree-constrained K-tree problem can be formulated as follows:

$$\begin{aligned}
\text{minimize} \quad & \sum_{i,j \in N} c_{ij} x_{ij} && && (11.26)\\
\text{subject to} \quad & \sum_{j \in N^-} x_{0j} = 2K && && (11.27)\\
& \sum_{\substack{i \in S, j \in \bar{S} \text{ or} \\ i \in \bar{S}, j \in S}} x_{ij} \geq 1 && \forall S \subseteq N^- : S \neq \emptyset && (11.28)\\
& x_{ij} \in \{0,1\} && \forall i,j \in N && (11.29)
\end{aligned}$$

Constraints (11.27) enforce the degree restriction on the depot, while constraints (11.28) ensure connectivity (at least one edge comes out of every subset). (This model prohibits single-customer routes, hence constraints (11.29) apply to the depot, as well.) This formulation can be seen as a relaxation of (11.1)–(11.6) by removing constraints (11.2), using (11.7) in place of (11.4) and replacing its right-hand side with 1 (which is always less than $2v(S)$). Fisher (1994b) shows that this problem can be solved in $O(n^3)$ time. Fisher (1994a) uses this lower bound in a branch-and-bound algorithm.

Unfortunately, neither the b-matching bound nor the K-tree bound is very tight, often falling 20% or more below the optimal VRP objective value (Toth and Vigo 2001b). Therefore, branch-and-bound methods based on these and other simple relaxations are generally not effective for any but the smallest problem instances.

11.2.3 Branch-and-Cut

Recall from Section 10.3.3 that a branch-and-cut algorithm involves relaxing certain constraints (integrality and/or functional constraints), solving the resulting problem, and then adding additional constraints ("cuts") that make the current optimal solution infeasible, thus tightening the formulation. One obvious choice for VRP constraints to relax is the capacity-cut constraints, which in this section we will assume are in the form given in (11.7), that is:

$$\sum_{\substack{i\in S, j\in \bar{S} \text{ or} \\ i\in \bar{S}, j\in S}} x_{ij} \geq 2v(S) \qquad \forall S \subseteq N^- : S \neq \emptyset.$$

As we discussed in Section 11.2.2, relaxing the capacity-cut constraints results in the b-matching problem, which can be solved efficiently. The question, then, is how to solve the separation problem—how to identify a violated inequality (a cut) that will render the solution to the relaxed problem infeasible.

The answer turns out to depend on how tight we wish the capacity-cut constraints to be. It is difficult to identify violated inequalities of the form (11.7), since calculating $v(S)$ itself is NP-hard due to its relationship to the bin-packing problem. Ralphs et al. (2003) propose a heuristic for solving the separation problem in this case. On the other hand, we noted in Section 11.1.3 that the formulation is still valid if we replace $v(S)$ by $\lceil d(S)/C \rceil$. The separation problem in this case is still difficult, but less so—it is still usually done heuristically. We can even replace $v(S)$ by the weaker value $d(S)/C$, which does not maintain the validity of the IP formulation but is more tractable. The separation problem for this form of the constraints can be solved in polynomial time (McCormick et al. 2003).

Since the VRP is a generalization of the TSP, any valid inequality developed for the TSP—for example, those discussed in Section 10.3.3—can be adapted for the VRP (Naddef and Rinaldi 1993). On the other hand, cuts derived in this way are often not particularly tight for the VRP, since these inequalities ignore the vehicle capacity. They can be strengthened by making use of the function $v(S)$—in essence, combining the capacity-cut inequalities (which account for the bin-packing aspect of the VRP) with the TSP-derived inequalities (which account for the routing aspect).

We illustrate this idea using comb inequalities. Recall from Section 10.3.3 that a *comb* consists of a set H called the *handle* and a collection of sets $T_1, \ldots, T_s$ called *teeth*, such that each T_j contains at least one node in H and one node not in H and such that s is odd and at least 3. A comb inequality for the TSP can be written as in (10.16), that is:

$$\sum_{\substack{i\in H \\ j\notin H}} x_{ij} + \sum_{k=1}^{s} \sum_{\substack{i\in T_k \\ j\notin T_k}} x_{ij} \geq 3s+1.$$

To adapt this for the VRP, Laporte and Nobert (1984) prove the following:

Theorem 11.2 *For any handle $H \subseteq N$ and teeth $T_1, \ldots, T_s \subseteq N$ such that*

- *no T_k contains the depot,*
- *each T_k contains at least one node in H and at least one node not in H,*
- *$T_1, \ldots, T_s$ are pairwise disjoint,*
- *$s \geq 3$ and odd, and*
- *$v(T_k \setminus H) + v(T_k \cap H) > v(T_k)$ for all $k = 1, \ldots, s$,*

the following inequality is valid for every VRP solution through N:

$$\sum_{\substack{i \in H \\ j \notin H}} x_{ij} + \sum_{k=1}^{s} \sum_{\substack{i \in T_k \\ j \notin T_k}} x_{ij} \geq s + 1 + 2\sum_{k=1}^{s} v(T_k). \tag{11.30}$$

Proof. Omitted. ∎

If $C = \infty$, then $v(T_k) = 1$ for all k, and (11.30) is identical to (10.16) for the TSP. Theorem 11.2 can be adapted to the case in which the depot is contained in one of the teeth, as well.

Another approach to adapting TSP comb inequalities for the VRP is as follows. Suppose we duplicate the depot so that there are K copies; call this set of depots D. Let $c_{ij} = \infty$ if $i, j \in D$. Then a TSP tour through $N^- \cup D$ is a feasible VRP solution if the total demand of the nodes between consecutive visits to depot nodes is no greater than C. A TSP comb inequality on $N^- \cup D$ can be converted to a VRP comb inequality by re-combining the nodes in D back into the single depot, but we must adapt the definition of a comb to deal with the fact that the teeth may now intersect (because multiple teeth may contain the depot). Assume that for some $1 \leq r \leq s$, teeth $T_1, \ldots, T_r$ do not intersect, and teeth $T_{r+1}, \ldots, T_s$ intersect only at the depot. Assume also that $r(N^- \setminus T_k) = K$ for all $k = r+1, \ldots, s$; in other words, if tooth T_k contains the depot, then the nodes not in the tooth require all K vehicles to serve them. Then one can show (see Problem 11.12) that the following inequality is valid for every VRP solution through N:

$$\sum_{\substack{i \in H \\ j \notin H}} x_{ij} + \sum_{k=1}^{s} \sum_{\substack{i \in T_k \\ j \notin T_k}} x_{ij} \geq s + 1 + 2r + 2K(s - r). \tag{11.31}$$

This inequality can be tighter than (11.30).

For a more thorough review of branch-and-cut approaches for the VRP, see Naddef and Rinaldi (2001).

11.2.4 Set Covering

Because VRP solutions consist of a set of disjoint routes, the VRP lends itself well to a set covering/column generation approach, similar to the approach in Section 12.2.7 for the LMRP and described further in Section D.2.4. This method was proposed by Balinski and Quandt (1964) and has since been refined by a number of other authors; see, e.g., Bramel and Simchi-Levi (2001) for an overview. (Another set of classical approaches, called *petal heuristics*, can be thought of as a simplified and approximate version of the set covering/column generation approach in which routes are generated heuristically; see, e.g., Foster and Ryan (1976) and Renaud et al. (1996).)

Suppose for a moment that we could enumerate all feasible routes, i.e., routes that begin and end at the depot, visit each customer at most once, and do not violate the capacity constraint. Let $\mathcal{R}$ be the set of all feasible routes; for a given route $r \in \mathcal{R}$, let c_r be the total length of the route; and let $a_{ir} = 1$ if i is on route r, 0 otherwise. Define a decision variable y_r, for $r \in \mathcal{R}$, as follows:

$$y_r = \begin{cases} 1, & \text{if route } r \text{ is in the solution,} \\ 0, & \text{otherwise.} \end{cases}$$

The set covering formulation for the VRP is as follows:

$$\text{(VRP-SC)} \quad \text{minimize} \quad \sum_{r \in \mathcal{R}} c_r y_r \tag{11.32}$$

$$\text{subject to} \quad \sum_{r \in \mathcal{R}} a_{ir} y_r \ge 1 \quad \forall i \in N^- \tag{11.33}$$

$$\sum_{r \in \mathcal{R}} y_r \le K \tag{11.34}$$

$$y_r \in \{0, 1\} \qquad \forall r \in \mathcal{R} \tag{11.35}$$

The objective function (11.32) calculates the total distance of the routes chosen. Constraints (11.33) require each node to be contained in a chosen route. Constraint (11.34) requires at most K routes to be used. (Here, we treat K as an upper bound on the number of routes.) Constraints (11.35) are integrality constraints. Note that, although constraints (11.33) are written as inequality constraints, they will always hold as equalities in the optimal solution. (Why?)

Of course, $\mathcal{R}$ is exponentially large, so it is not practical to enumerate all feasible routes for even moderately sized instances. Therefore, the set covering algorithm begins by enumerating only a (relatively small) subset $\mathcal{R}' \subseteq \mathcal{R}$ of feasible routes. This can be done randomly, or using some heuristic. Let (VRP-SC$'$) be the problem (VRP-SC) restricted to $\mathcal{R}'$, and let ($\overline{\text{VRP-SC}'}$) be its LP relaxation. Since $\mathcal{R}'$ is not too large, ($\overline{\text{VRP-SC}'}$) is relatively easy to solve using a standard LP solver. Let $\bar{y}$ be the optimal solution to ($\overline{\text{VRP-SC}'}$). How can we tell whether $\bar{y}$ is optimal for ($\overline{\text{VRP-SC}}$) (the LP relaxation of the original problem, with the full set $\mathcal{R}$)?

To answer this question, we'll formulate the dual of ($\overline{\text{VRP-SC}}$), letting π be the dual variables corresponding to constraints (11.33) and μ be the dual variable for (11.34):

$$(\overline{\text{VRP-SC-D}}) \quad \text{maximize} \quad \sum_{i \in N^-} \pi_i - K\mu \tag{11.36}$$

$$\text{subject to} \quad \sum_{i \in N^-} a_{ir} \pi_i - \mu \le c_r \quad \forall r \in \mathcal{R} \tag{11.37}$$

$$\pi_i \ge 0 \qquad \forall i \in N^- \tag{11.38}$$

$$\mu \ge 0 \tag{11.39}$$

Let $(\bar{\pi}, \bar{\mu})$ be the optimal dual values corresponding to the optimal primal solution $\bar{y}$ for ($\overline{\text{VRP-SC}'}$). If $(\bar{\pi}, \bar{\mu})$ is feasible for ($\overline{\text{VRP-SC-D}}$), then it is optimal for ($\overline{\text{VRP-SC-D}}$)

(why?) and $\bar{y}$ is optimal for ($\overline{\text{VRP-SC}}$). Thus, checking optimality of $\bar{y}$ for ($\overline{\text{VRP-SC}}$) is equivalent to checking feasibility of $(\bar{\pi}, \bar{\mu})$ for ($\overline{\text{VRP-SC-D}}$). But checking feasibility is not straightforward, since ($\overline{\text{VRP-SC-D}}$) has an exponential number of constraints, most of which we have not enumerated.

The solution to this challenge is to look explicitly for a violated constraint, i.e., for an $r \in \mathcal{R}$ such that

$$\sum_{i \in N^-} a_{ir}\bar{\pi}_i > c_r + \bar{\mu}, \tag{11.40}$$

or, equivalently, such that $\bar{c}_r > 0$, where

$$\bar{c}_r = c_r + \bar{\mu} - \sum_{i \in N^-} a_{ir}\bar{\pi}_i$$

is the *reduced cost* of column r. In other words, we want to solve the following *column generation problem*:

$$\text{(VRP-CG)} \quad \text{minimize} \quad \bar{c}_r \tag{11.41}$$

$$\text{subject to} \quad \sum_{i \in N^-} a_{ir} d_i \leq C \tag{11.42}$$

This problem searches for the tour $r \in \mathcal{R}$ that minimizes $\bar{c}_r$. If the optimal $\bar{c}_r$ is negative, then we have found a constraint that $(\bar{\pi}, \bar{\mu})$ violates, and moreover, we have found a new column that we should add to $\mathcal{R}'$. If the optimal $\bar{c}_r$ is nonnegative, then we have proven that $\bar{y}$ is optimal for ($\overline{\text{VRP-SC}}$).

Now the question is how to solve (VRP-CG). This problem itself is NP-hard, because for a given route r, even evaluating $\bar{c}_r$ requires finding an optimal TSP tour through the nodes on the route. It is usually solved using branch-and-bound or branch-and-cut (Agarwal et al. 1989, Desrochers et al. 1992, Bixby 1998).

Even after doing all of this, we have still only solved the LP relaxation of (VRP-SC). To solve (VRP-SC) itself, one approach is to use $\mathcal{R}'$ as a starting point in a branch-and-price algorithm to solve (VRP-SC) exactly. In this approach, new columns are generated as needed within the branch-and-bound tree, with the net result of solving (VRP-SC) without enumerating all of the columns in $\mathcal{R}$ explicitly. See Desrochers et al. (1992) for an algorithm of this type.

Another approach is to solve (VRP-SC$'$) exactly with the current set of columns, i.e., solving the VRP restricted to the routes in $\mathcal{R}'$, by branch-and-cut or another method (see, e.g., Bramel and Simchi-Levi (2001)). Since we are not solving the full problem to optimality, this approach is a heuristic.

Both methods are quite effective, in large part due to the fact that the LP relaxation of the set covering problem tends to be very tight; in fact, it often has all-integer solutions. (We made a similar observation about the uncapacitated fixed-charge location problem (UFLP) on page 272.) This has been observed empirically (e.g., Desrochers et al. 1992), and in fact Simchi-Levi et al. (2013) prove that the LP bound for the set covering formulation of the VRP approaches the IP value asymptotically as $n \to \infty$. Bramel and Simchi-Levi (1997) prove this for the more general VRP with time windows.

The set covering/column generation algorithm is summarized in Algorithm 11.1.

Algorithm 11.1 Set covering-based algorithm for VRP

$\mathcal{R}' \leftarrow$ partial enumeration of $\mathcal{R}$ $\triangleright$ Initialization
repeat $\triangleright$ Solve LP relaxation
  $(\bar{y}, \bar{\pi}, \bar{\mu}) \leftarrow$ opt. primal, dual solution to $(\overline{\text{VRP-SC}'})$ $\triangleright$ Solve restricted LP
  $r^* \leftarrow$ opt. solution to (VRP-CG) with $(\bar{\pi}, \bar{\mu})$ $\triangleright$ Search for new column
  if $\bar{c}_{r^*} < 0$ **then**
    $\mathcal{R}' \leftarrow \mathcal{R}' \cup \{r^*\}$ $\triangleright$ Add new column
  end if
until $\bar{c}_{r^*} \geq 0$ $\triangleright$ $\bar{y}$ is now optimal for $(\overline{\text{VRP-SC}})$
use $\mathcal{R}'$ to find exact (or approximate) optimal IP solution $\triangleright$ Solve IP
return IP solution

11.3 HEURISTICS FOR THE VRP

The VRP is a particularly difficult combinatorial optimization problem, because of the need to make simultaneous decisions about both clustering and routing. Moreover, the LP relaxation bounds from the IP formulations in Section 11.1.3 are not particularly tight, and it is more difficult to derive strong lower bounds in other ways; therefore, pure branch-and-bound algorithms do not tend to have acceptable performance, and branch-and-cut approaches for the VRP have not yet caught up to those for the TSP. For these reasons, heuristics are of particular importance for the VRP.

In this section, we will discuss several construction heuristics for the VRP. We will discuss improvement heuristics in Section 11.3.4, but we will spend relatively less effort on these methods since many of the improvement heuristics for the TSP can also be applied directly to the VRP.

11.3.1 The Clarke–Wright Savings Heuristic

The *Clarke–Wright savings heuristic* (Clarke and Wright 1964) is one of the best-known heuristics for the VRP. The heuristic assumes that the number of vehicles is unrestricted. It begins by placing each node on its own route and then merging routes when doing so reduces the total distance.

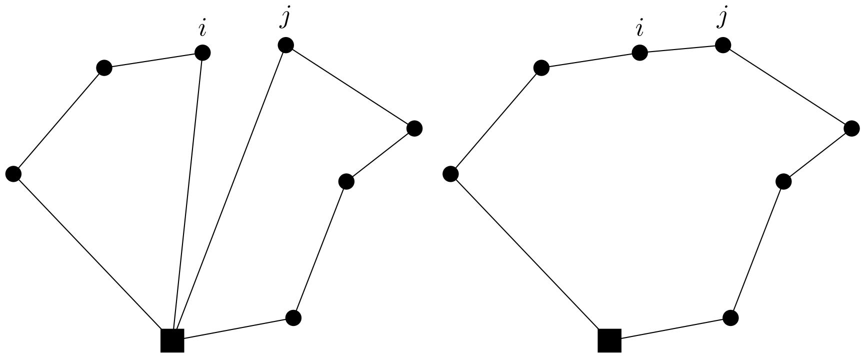
Consider the two routes shown in Figure 11.2(a). Suppose we were to merge the two routes by adding an edge from node i to node j, as shown in Figure 11.2(b). The *savings* from such a merger is given by

$$s_{ij} = c_{i0} + c_{0j} - c_{ij}. \tag{11.43}$$

By the triangle inequality, $s_{ij} > 0$.

The savings heuristic builds a *savings list* of the s_{ij} values for all $i \neq j \in N^-$, sorted in descending order. The algorithm then proceeds down the list, implementing, at each iteration, a merger of two routes at nodes i and j if the following conditions hold:

1. i and j are on different routes.
2. i and j are both adjacent to the depot on their respective routes.
3. The resulting route would be feasible with respect to capacity constraints.



(a) Routes before merger. (b) Route after merger.

Figure 11.2 Clarke–Wright savings heuristic.

Once a merger has been considered and either implemented or rejected, it is never considered again, since none of the conditions above change from false to true during the course of the heuristic. The algorithm terminates when every merger has been considered. Pseudocode for the heuristic is given in Algorithm 11.2.

Algorithm 11.2 Clarke–Wright savings heuristic

for all $i, j \in N^-, i \neq j$ **do** ▷ Build savings list
    $s_{ij} \leftarrow c_{i0} + c_{0j} - c_{ij}$
end for
sort s_{ij} in decreasing order
for all $i \in N^-$ **do** ▷ Initial routes
    place i on its own route
end for
for all $i, j \in N^-$ **do** ▷ Route mergers
    if $s_{ij} > 0$, i and j are on different routes, i and j are adjacent to node 0, and combined demand of routes containing i and j is $\leq C$ **then**
        merge routes at i and j
    end if
end for
return routes

☐ **EXAMPLE 11.1**

Consider the instance pictured in Figure 11.3. Each customer's demand is noted next to it. The vehicle capacity is $C = 5$. Distances c_{ij} are Euclidean. The complete

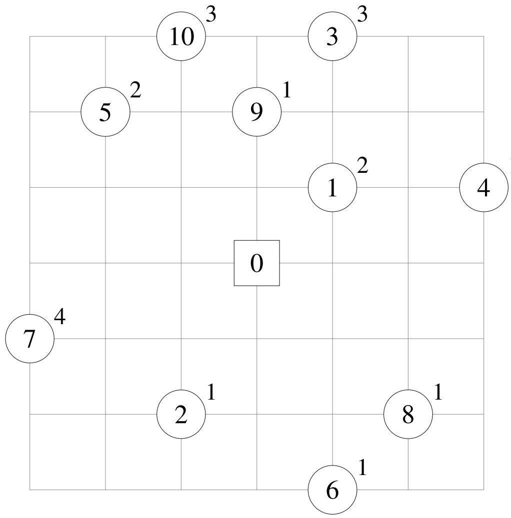


Figure 11.3 VRP instance for examples. Distances are Euclidean.

distance matrix is given by

$$c = \begin{bmatrix} 0.00 & 1.41 & 2.24 & 3.16 & 3.16 & 2.83 & 3.16 & 3.16 & 2.83 & 2.00 & 3.16 \\ 1.41 & 0.00 & 3.61 & 2.00 & 2.00 & 3.16 & 4.00 & 4.47 & 3.16 & 1.41 & 2.83 \\ 2.24 & 3.61 & 0.00 & 5.39 & 5.00 & 4.12 & 2.24 & 2.24 & 3.00 & 4.12 & 5.00 \\ 3.16 & 2.00 & 5.39 & 0.00 & 2.83 & 3.16 & 6.00 & 5.66 & 5.10 & 1.41 & 2.00 \\ 3.16 & 2.00 & 5.00 & 2.83 & 0.00 & 5.10 & 4.47 & 6.32 & 3.16 & 3.16 & 4.47 \\ 2.83 & 3.16 & 4.12 & 3.16 & 5.10 & 0.00 & 5.83 & 3.16 & 5.66 & 2.00 & 1.41 \\ 3.16 & 4.00 & 2.24 & 6.00 & 4.47 & 5.83 & 0.00 & 4.47 & 1.41 & 5.10 & 6.32 \\ 3.16 & 4.47 & 2.24 & 5.66 & 6.32 & 3.16 & 4.47 & 0.00 & 5.10 & 4.24 & 4.47 \\ 2.83 & 3.16 & 3.00 & 5.10 & 3.16 & 5.66 & 1.41 & 5.10 & 0.00 & 4.47 & 5.83 \\ 2.00 & 1.41 & 4.12 & 1.41 & 3.16 & 2.00 & 5.10 & 4.24 & 4.47 & 0.00 & 1.41 \\ 3.16 & 2.83 & 5.00 & 2.00 & 4.47 & 1.41 & 6.32 & 4.47 & 5.83 & 1.41 & 0.00 \end{bmatrix} \tag{11.44}$$

The positive entries of the sorted savings list are given in Table 11.1. For example:

$$\begin{aligned} s_{6,8} &= c_{6,0} + c_{0,8} - c_{6,8} = 3.16 + 2.83 - 1.41 = 4.58 \\ s_{5,10} &= c_{5,0} + c_{0,10} - c_{5,10} = 2.83 + 3.16 - 1.41 = 4.58 \\ s_{3,10} &= c_{3,0} + c_{0,10} - c_{3,10} = 3.16 + 3.16 - 2.00 = 4.32, \end{aligned}$$

and so on.

We begin by placing each customer on its own route (Figure 11.4(a)). Next, we consider the first pair on the savings list, $(6, 8)$. Merging the routes containing 6 and 8 is feasible with respect to the conditions on page 475, so we merge them (Figure 11.4(b)). The next pair—$(5, 10)$—is feasible as well, so we merge the two routes (Figure 11.4(c)). The routes containing 3 and 10 cannot be merged because their total demand is 8, which exceeds the capacity, so we skip that entry in the savings list. The merger $(3, 9)$ is feasible (Figure 11.4(d)); $(9, 10)$ is not; $(3, 4)$ is feasible (Figure 11.4(e)), as is $(2, 6)$ (Figure 11.4(f)). The next several mergers are

Table 11.1 All positive entries of sorted savings list for VRP instance in Figure 11.3.

s_{ij}	i	j	s_{ij}	i	j	s_{ij}	i	j	s_{ij}	i	j
4.58	6	8	2.83	5	9	1.08	1	5	0.36	8	9
4.58	5	10	2.58	1	3	1.08	1	8	0.32	3	6
4.32	3	10	2.58	1	4	0.94	2	5	0.16	5	6
3.75	3	9	2.06	2	8	0.92	7	9	0.16	8	10
3.75	9	10	2	4	9	0.89	4	5	0.11	2	9
3.5	3	4	2	1	9	0.89	3	8	0.1	1	7
3.16	2	6	1.85	4	6	0.89	7	8	0.06	6	9
3.16	2	7	1.85	6	7	0.67	3	7	0.04	1	2
2.83	3	5	1.85	4	10	0.58	1	6	0.01	2	3
2.83	5	7	1.85	7	10	0.4	2	4		—	
2.83	4	8	1.75	1	10	0.4	2	10		—	

all infeasible, until we get to $(1, 8)$. The remaining mergers are all infeasible or have 0 savings. The solution returned by the heuristic is shown in Figure 11.4(g); it has a total distance of 33.60. □

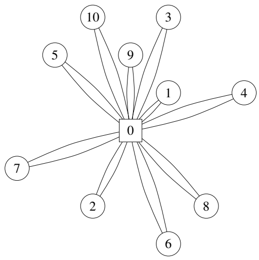
The Clarke–Wright solution to the `eil51` instance in Figure 11.1 has a total distance of 582 (compared to the optimal distance of 521) and is pictured in Figure 11.5.

A variant of the savings heuristic skips the sorting step (line 4) and instead implements mergers with positive savings in the order in which they are found. Laporte and Semet (2001) refer to this variant and the version in Algorithm 11.2 as the *sequential* and *parallel* versions, respectively. (Alternate terms might be *first-improving* and *best-improving*.) They report that the parallel version outperforms the sequential version considerably and warn that some authors neglect to indicate which version they are using when reporting computational results.

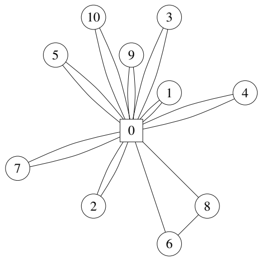
On the other hand, it is sometimes worthwhile to consider mergers other than the one with the largest savings through a *randomization* mechanism. For example, we might choose randomly from among the ℓ best feasible mergers to obtain a solution; we can repeat this several times, possibly with different values of ℓ, and choose the best solution found. Daskin (2010) proposes an approach like the one listed in Algorithm 11.3. L is called the *randomization depth* and M is called the *randomization iterations*; both are inputs to the algorithm and are typically small, say, $L = M = 5$.

Algorithm 11.3 Randomized Clarke–Wright savings heuristic

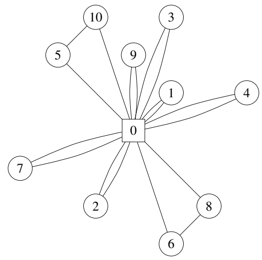
for $\ell = 1, \ldots, L$ **do**
  for $m = 1, \ldots, M$ **do**
    run Clarke–Wright savings heuristic in Algorithm 11.2, but at step 10, choose randomly from among the ℓ feasible mergers with greatest savings
  end for
end for
return solution found with smallest total distance



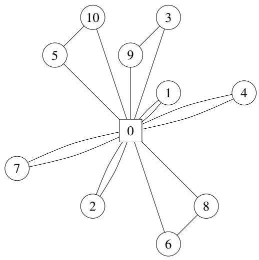
(a) Each customer on own route.



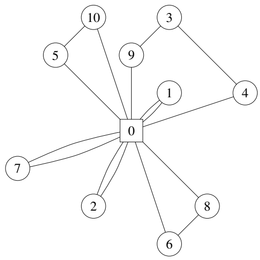
(b) After merger $(6, 8)$.



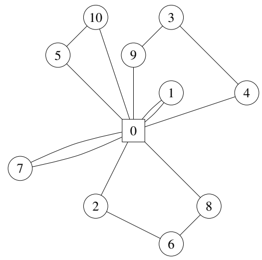
(c) After merger $(5, 10)$.



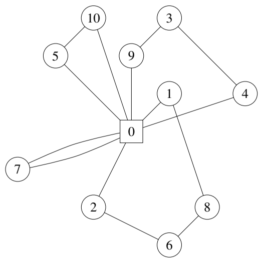
(d) After merger $(3, 9)$.



(e) After merger $(3, 4)$.



(f) After merger $(2, 6)$.



(g) After merger $(1, 8)$—final result. Total distance = 33.60.

Figure 11.4 Clarke–Wright savings heuristic for instance in Figure 11.1.

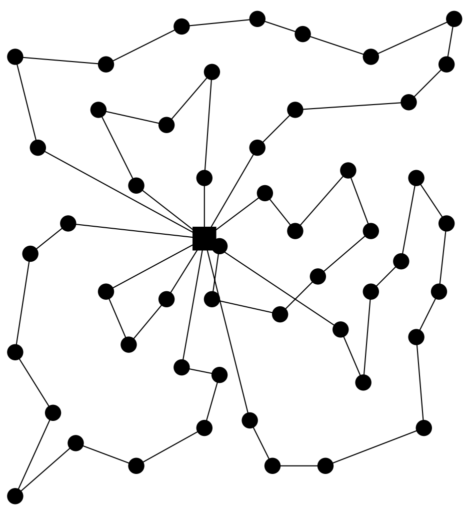


Figure 11.5 Clarke–Wright solution to `eil51` VRP instance. Total distance = 582.

Sometimes route mergers are selected using a matching algorithm; see, e.g., Desrochers and Verhoog (1989) and Wark and Holt (1994). These methods tend to be a bit more accurate than the classical Clarke–Wright algorithm but also significantly slower.

Another variant, proposed by Gaskell (1967) and Yellow (1970), uses a generalized savings calculation,

$$s_{ij} = c_{i0} + c_{0j} - \lambda c_{ij},$$

where λ is a route shape parameter that allows the user to specify how much emphasis to place on the distance between the two nodes to be merged. If λ is large, mergers are penalized if i and j are far from each other. This tends to encourage more compact routes.

11.3.2 The Sweep Heuristic

The *sweep heuristic* (Wren 1971, Wren and Holliday 1972, Gillett and Miller 1974) builds clusters of nodes by rotating a ray emanating from the depot, adding nodes as the ray hits them, and beginning a new cluster when the next node would violate the vehicle capacity. Routes are then constructed by solving a TSP (exactly or heuristically) for each cluster. Typically, the number of routes is unrestricted. In the pseudocode for the sweep heuristic in Algorithm 11.4, (α_i, ρ_i) represents the polar coordinates of node $i \in N$, and S_k represents cluster k of nodes.

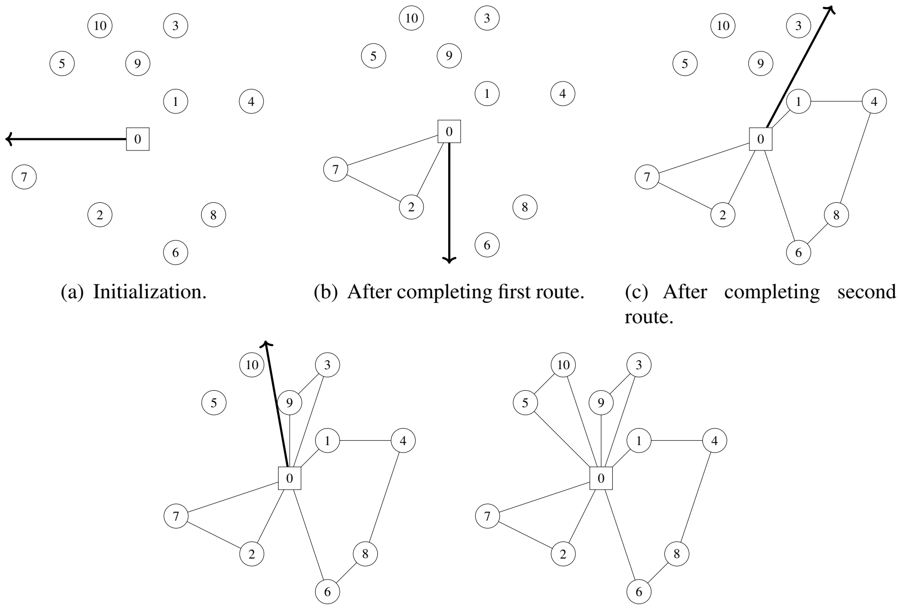
Algorithm 11.4 Sweep heuristic

$\alpha \leftarrow$ arbitrary angle; $k \leftarrow 1$; $S_k \leftarrow \emptyset\ \forall k$ ▷ Initialization
while some nodes are not in any cluster **do** ▷ Clustering
    increase α until it equals α_i for some i not in a cluster
    if $\sum_{j \in S_k} d_j + d_i > C$ **then**
        $k \leftarrow k + 1$
    end if
    $S_k \leftarrow S_k \cup \{i\}$ ▷ Update cluster
end while
for all k **do** ▷ Route optimization
    solve TSP on nodes in S_k
end for
return routes

Routes produced by the sweep heuristic never overlap, but optimal routes often do. Therefore, node-exchange improvement heuristics (see Section 11.3.4) can be particularly useful after the heuristic completes its execution.

The sweep heuristic is an example of a *two-phase method* in which clustering and routing are done in two separate steps. Two-phase methods come in two types: cluster-first, route-second and route-first, cluster-second. The sweep method is an example of the former type,[3] as is the location-based heuristic discussed in Section 11.3.3. In contrast, route-first, cluster-second methods solve a TSP on the entire node set and then partition the tour into routes (Beasley 1983). Laporte and Semet (2001) observe that these methods

[3]Simchi-Levi et al. (2013) disagree with this categorization, classifying the sweep heuristic as a route-first, cluster-second method since clustering is done on a fixed ordering of the nodes.



(a) Initialization.

(b) After completing first route.

(c) After completing second route.

(d) After completing third route.

(e) Final result. Total distance = 32.77.

Figure 11.6 Sweep heuristic for instance in Figure 11.1.

rarely perform better than cluster-first, route-second approaches, and Li and Simchi-Levi (1990) provide a theoretical justification.

☐ **EXAMPLE 11.2**

Return to the instance pictured in Figure 11.3. Let's start the sweep heuristic at $\alpha = \pi$ (Figure 11.6(a)) and sweep counter-clockwise. We first hit node 7, then node 2, bringing the load of the first route to 5. Obviously, we cannot add node 6 to the same route, so we start a new one, which can also accommodate 8 (Figure 11.6(b)). The next four nodes—6, 8, 4, and 1—can all be added to a single route, so we do (Figure 11.6(c)). Nodes 3 and 9 fit on a route, leaving one unit of capacity, but since the demand of node 10 is 3, we cannot put it on the same route (Figure 11.6(d)). The final route consists of nodes 10 and 5, and the resulting soltuion has a total distance of 32.77 (Figure 11.6(e)). These routes are each already optimal TSP tours, so no further optimization is performed. ☐

For the `eil51` instance in Figure 11.1, the sweep heuristic returns the solution pictured in Figure 11.7, which has a total distance of 586.27. (The routes were optimized using the farthest insertion heuristic rather than with an exact algorithm.)

11.3.3 The Location-Based Heuristic

The *location-based heuristic* (LBH) of Bramel and Simchi-Levi (1995) approximates the VRP by the *capacitated concentrator location problem* (CCLP), a close variant of the ca-

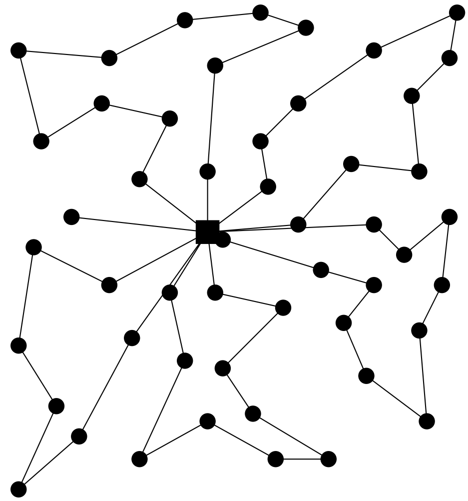


Figure 11.7 Sweep heuristic solution to `eil51` VRP instance. Total distance = 586.27.

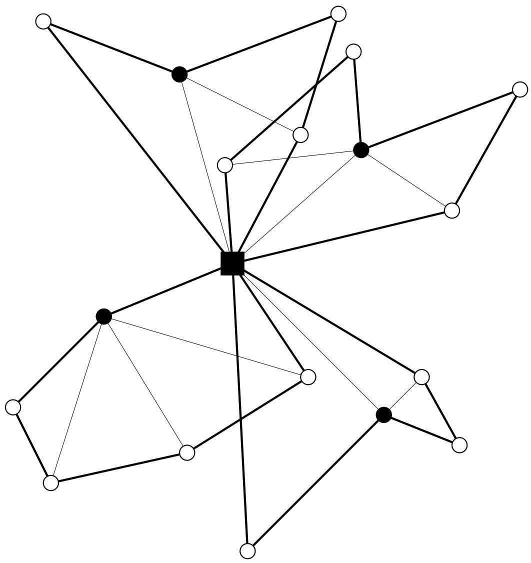


Figure 11.8 Approximating the VRP (thick lines) by the CCLP (thin lines) in the location-based heuristic. ■ = depot, ● = seed node in CCLP, and ○ = non-seed node in CCLP.

pacitated facility location problem (CFLP) with single-sourcing constraints (Section 8.3.1) in which the demands are ignored in the transportation costs but not in the capacity constraints. The basic idea is to use the CCLP to cluster the nodes and then solve TSPs to optimize the individual routes; see Figure 11.8. The facilities opened in the CCLP solution are called "seed nodes." The choice of seed nodes is actually rather inconsequential; what is important is the cluster of nodes that are assigned to each seed node (including the seed node itself), since these will form the node sets for the individual routes. The heuristic assumes that the number of routes is unrestricted.

We will assume that the sets of customers and potential facility locations (called I and J in the CFLP) are both equal to N^-. We would like the cost of locating a facility at node j and serving a set S_j of customer nodes in the CCLP to approximate the cost (length) of a TSP tour through the depot and the nodes in S_j. The LBH divides the cost of this tour into two components, the portion to and from the depot and the portion among the other nodes. It includes the length of the former portion in the fixed location cost f_j and that of the latter portion in the transportation costs $\tilde{c}_{ij}$.[4] In particular, for each $j \in N^-$, we set

$$f_j = 2c_{0j}, \tag{11.45}$$

and for each $i, j \in N^-$, we set $\tilde{c}_{ij}$ as either

$$\tilde{c}_{ij} = 2c_{ij} \tag{11.46}$$

or

$$\tilde{c}_{ij} = c_{0i} + c_{ij} - c_{j0}. \tag{11.47}$$

If we locate a facility at node j, then (11.46) approximates node i's contribution to the TSP tour length as the distance from j to i and back, while (11.47) approximates it using the cost to insert node i into the tour that goes from the depot to node j and back. Bramel and Simchi-Levi (1995) refer to (11.46) as the *star connection* cost and to (11.47) as the *seed tour* cost. We'll use LBH-SC and LBH-ST to refer to the LBH heuristic using these two costs, respectively. Neither is meant to model the TSP tour cost exactly; the aim is simply to find costs for the CCLP that tend to produce solutions that translate to good solutions for the VRP. The computational tests by Bramel and Simchi-Levi (1995) suggest that LBH-ST performs somewhat better computationally, but LBH-SC has nice theoretical properties; see Theorem 11.5 below.

Given these costs, we can formulate the CCLP:

$$
\begin{aligned}
\text{(CCLP)} \quad \text{minimize} \quad & \sum_{j \in N^-} f_j x_j + \sum_{i \in N^-} \sum_{j \in N^-} \tilde{c}_{ij} y_{ij} && && (11.48)\\
\text{subject to} \quad & \sum_{j \in N^-} y_{ij} = 1 && \forall i \in N^- && (11.49)\\
& y_{ij} \le x_j && \forall i \in N^-, \forall j \in N^- && (11.50)\\
& \sum_{i \in N^-} d_i y_{ij} \le C && \forall j \in N^- && (11.51)\\
& x_j \in \{0,1\} && \forall j \in N^- && (11.52)\\
& y_{ij} \in \{0,1\} && \forall i \in N^-, \forall j \in N^- && (11.53)
\end{aligned}
$$

(CCLP) can be solved using any available method; Bramel and Simchi-Levi (1995) suggest using the Lagrangian relaxation method described in Section 8.3.1, calculating β_j by solving a 0–1 knapsack problem. To find feasible (upper-bound) solutions to the CCLP, they open facilities in order of β_j, and for each new facility, they assign customers by solving a new knapsack problem on the customers that have not yet been assigned.

[4]We will denote the CCLP transportation costs as $\tilde{c}_{ij}$ in this chapter to distinguish them from the VRP distances c_{ij}.

Once we have solved the CCLP, we construct clusters of nodes, each of which consists of the nodes assigned to a given seed node in the CCLP solution. We then solve a TSP on the nodes in each cluster, either exactly or approximately. The LBH is summarized in Algorithm 11.5.

Algorithm 11.5 Location-based heuristic for VRP

$f_j \leftarrow 2c_{0j}$, $\tilde{c}_{ij} \leftarrow 2c_{ij}$ or $c_{0i} + c_{ij} - c_{j0}$ $\forall i, j$ ▷ CCLP instance
solve CCLP with costs f_j, $\tilde{c}_{ij}$
for all $j \in N^-$ s.t. $x_j = 1$ in CCLP solution **do** ▷ Clustering
    $S_j \leftarrow \{i \in N^- | y_{ij} = 1$ in CCLP solution$\}$
end for
for all $j \in N^-$ s.t. $x_j = 1$ in CCLP solution **do** ▷ Routing
    solve TSP on nodes in S_j
end for
return routes

An earlier heuristic, by Fisher and Jaikumar (1981), is similar in spirit to the LBH, but instead of clustering via a facility location problem, it does so by solving a generalized assignment problem. Fisher and Jaikumar (1981) report good computational results for their algorithm, but the results have been difficult to replicate (Cordeau et al. 2002).

☐ **EXAMPLE 11.3**

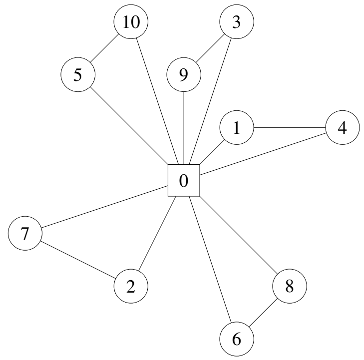
Return to the instance pictured in Figure 11.3. For the CCLP, the fixed costs f_j and star connection costs are calculated using (11.45) and (11.46) with the distance matrix given in (11.44). The solution to the resulting CCLP has $x_1 = x_2 = x_5 = x_8 = x_9 = 1$ (and the remaining $x_j = 0$), and $y_{39} = y_{41} = y_{68} = y_{72} = y_{10,5} = 1$, in addition to $y_{jj} = 1$ for all j such that $x_j = 1$. The resulting VRP solution is pictured in Figure 11.9(a) and has total distance 35.60.

Now suppose we use the seed tour costs:

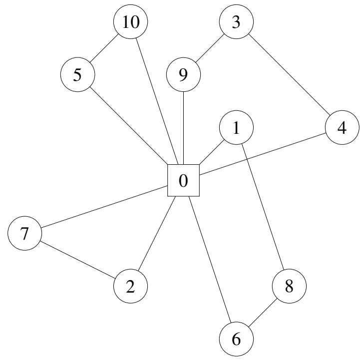
$$\tilde{c} = \begin{bmatrix} 0.00 & 2.78 & 0.25 & 0.25 & 1.75 & 2.25 & 2.72 & 1.75 & 0.83 & 1.08 \\ 4.43 & 0.00 & 4.46 & 4.07 & 3.53 & 1.31 & 1.31 & 2.41 & 4.36 & 4.07 \\ 3.75 & 6.31 & 0.00 & 2.83 & 3.50 & 6.00 & 5.66 & 5.43 & 2.58 & 2.00 \\ 3.75 & 5.93 & 2.83 & 0.00 & 5.43 & 4.47 & 6.32 & 3.50 & 4.32 & 4.47 \\ 4.58 & 4.72 & 2.83 & 4.77 & 0.00 & 5.50 & 2.83 & 5.66 & 2.83 & 1.08 \\ 5.75 & 3.16 & 6.00 & 4.47 & 6.16 & 0.00 & 4.47 & 1.75 & 6.26 & 6.32 \\ 6.22 & 3.16 & 5.66 & 6.32 & 3.50 & 4.47 & 0.00 & 5.43 & 5.40 & 4.47 \\ 4.58 & 3.59 & 4.77 & 2.83 & 5.66 & 1.08 & 4.77 & 0.00 & 5.30 & 5.50 \\ 2.00 & 3.89 & 0.25 & 2.00 & 1.17 & 3.94 & 3.08 & 3.64 & 0.00 & 0.25 \\ 4.58 & 5.93 & 2.00 & 4.47 & 1.75 & 6.32 & 4.47 & 6.16 & 2.58 & 0.00 \end{bmatrix}$$

The solution to the resulting CCLP has $x_2 = x_3 = x_8 = x_{10} = 1$ and $y_{18} = y_{43} = y_{5,10} = y_{68} = y_{72} = y_{93} = 1$. The resulting VRP solution is pictured in Figure 11.9(b) and has total distance 33.60. ☐

The LBH-SC heuristic is *asymptotically optimal* as the number of nodes increases. In other words, as $n \to \infty$, the total distance of the solution returned by the LBH-SC heuristic approaches the optimal VRP distance.



(a) With star connection costs. Total distance = 35.60.



(b) With seed tour connection costs. Total distance = 33.60.

Figure 11.9 LBH solution for instance in Figure 11.1.

Before proving the asymptotic optimality of LBH-SC, we return to the connection between the VRP and the bin-packing problem. Recall that, in the bin-packing problem, we are given a set of objects, each with a given weight (or other measure of size). The objective is to "pack" the objects into bins, each of which has a fixed capacity, minimizing the total number of bins used. Let b_n^* be the minimum number of bins of capacity C that are needed to pack n objects whose weights are drawn from a given probability distribution. It is well known that b_n^* converges to a constant as n increases:

$$\lim_{n\to\infty} \frac{b_n^*}{n} = \gamma,$$

where γ is a constant (known as the *bin-packing constant*) that depends on the probability distribution of the weights.

Suppose the demands d_i in the VRP are drawn from a given probability distribution. Then the problem of minimizing the number of trucks of capacity C required to serve n nodes (ignoring the routing aspect) is equivalent to the bin-packing problem, and the minimum required number of trucks is b_n^*, which can be approximated by $n\gamma$ for large enough n. Bramel et al. (1992) use this fact to characterize the asymptotic behavior of the optimal VRP objective function value:

Theorem 11.3 *Suppose that the node locations in the VRP are drawn iid from a probability distribution on a compact region with expected distance $\mathbb{E}[c]$ to the depot. Suppose that d_i/C (the node demand divided by the vehicle capacity) is drawn iid from a probability distribution F with support on $[0,1]$. Let z_n^* be the optimal objective function value for the problem with n nodes. Then*

$$\lim_{n\to\infty} \frac{1}{n} z_n^* = 2\gamma\mathbb{E}[c] \tag{11.54}$$

almost surely (a.s.), where γ is the bin-packing constant for distribution F.

Proof. Omitted; see Bramel et al. (1992). ■

In other words, for sufficiently large n, z_n^* can be approximated by $2n\gamma\mathbb{E}[c]$, the cost of using $n\gamma$ vehicles and sending each to a node at a distance of $\mathbb{E}[c]$ from the depot. This approximate value depends on γ (which, in turn, depends on the demand distribution and vehicle capacity) and the expected distance from the depot to the nodes.

Let z_n^{SC} be the total distance of the VRP solution returned by LBH-SC. First note that this distance is bounded above by z_n^L, the optimal CCLP cost:

Lemma 11.4 *For a VRP instance with n nodes,*

$$z_n^{SC} \le z_n^L.$$

Proof. Omitted; see Problem 11.20. ∎

We are now ready to prove that the LBH-SC heuristic is asymptotically optimal.

Theorem 11.5 *Under the same conditions as in Theorem 11.3,*

$$\lim_{n\to\infty} \frac{1}{n} z_n^{SC} = 2\gamma \mathbb{E}[c] \tag{11.55}$$

a.s.; i.e., the LBH-SC heuristic is asymptotically optimal.

Proof. We first prove that

$$\lim_{n\to\infty} \frac{1}{n} z_n^L = 2\gamma \mathbb{E}[c].$$

We do this by constructing a family of feasible solutions to the CCLP (parameterized by a constant $\epsilon > 0$) whose cost divided by n approaches $2\gamma\mathbb{E}[c]$.

Let $\epsilon > 0$ be a constant. Overlay a grid whose squares have side length ϵ atop the region $A \subseteq \mathbb{R}^2$ in which the nodes are located, and let $\{A_m\}_{m=1}^M$ be the subregions of A defined by this grid. (See Figure 11.10(a).) Let $n(m)$ be the number of nodes located in subregion A_m. Let $b^*(m)$ be the minimum number of bins of capacity C required to pack the $n(m)$ nodes in subregion A_m. Let $N_k(m) \subseteq N^-$ be the set of nodes in the kth bin of this packing and let i_k be an arbitrarily selected node in $N_k(m)$, for $k = 1, \ldots, b^*(m)$. (See Figure 11.10(b). Note that the nodes in a given bin need not be clustered geographically, though we have drawn them in that way for simplicity.)

Construct a feasible solution to the CCLP by establishing a facility (concentrator) at node i_k and assigning the remaining nodes in $N_k(m)$ to i_k, for $k = 1, \ldots, b^*(m)$. By (11.45) and (11.46), this solution has cost

$$\sum_{m=1}^{M} \sum_{k=1}^{b^*(m)} \left(2c_{0i_k} + \sum_{j\in N_k(m)} 2c_{i_k j} \right),$$

and the optimal solution has a cost z_n^L that is less than or equal to this cost. Since A_m is contained in a square of side length ϵ, $c_{i_k j} \le \epsilon\sqrt{2}$ for all $j \in N_k(m) \setminus \{i_k\}$, and $c_{i_k i_k} = 0$, so

$$\begin{aligned} z_n^L &\le 2 \sum_{m=1}^{M} \sum_{k=1}^{b^*(m)} \left(c_{0i_k} + \sum_{j\in N_k(m)\setminus\{i_k\}} \epsilon\sqrt{2} \right) \\ &= 2 \sum_{m=1}^{M} \sum_{k=1}^{b^*(m)} \left(c_{0i_k} + (|N_k(m)| - 1)\epsilon\sqrt{2} \right). \end{aligned}$$

Let $c^*(m)$ be the distance from the depot to the nearest node in A_m. By the triangle inequality, $c_{0i_k} \leq c^*(m) + \epsilon\sqrt{2}$ for all $m = 1, \ldots, M$, so

$$\begin{aligned} z_n^L &\leq 2 \sum_{m=1}^{M} \sum_{k=1}^{b^*(m)} \left(c^*(m) + \epsilon\sqrt{2} + (|N_k(m)| - 1)\epsilon\sqrt{2} \right) \\ &= 2 \sum_{m=1}^{M} \left(b^*(m)(c^*(m) + \epsilon\sqrt{2}) + \sum_{k=1}^{b^*(m)} (|N_k(m)| - 1)\epsilon\sqrt{2} \right) \\ &= 2 \sum_{m=1}^{M} b^*(m)c^*(m) + 2n\epsilon\sqrt{2} \end{aligned}$$

since $\bigcup_{m=1}^{M} \bigcup_{k=1}^{b^*(m)} N_k(m) = N^-$. Therefore,

$$\limsup_{n\to\infty} \frac{1}{n} z_n^L \leq 2 \limsup_{n\to\infty} \frac{1}{n} \sum_{m=1}^{M} b^*(m)c^*(m) + 2\epsilon\sqrt{2}.$$

(We need $\limsup$ rather than $\lim$ because of the inequality.) Bramel et al. (1992) prove that

$$\limsup_{n\to\infty} \frac{1}{n} \sum_{m=1}^{M} b^*(m)c^*(m) \leq \gamma\mathbb{E}[c]$$

a.s. Therefore,

$$\limsup_{n\to\infty} \frac{1}{n} z_n^L \leq 2\left(\gamma\mathbb{E}[c] + \epsilon\sqrt{2} \right)$$

a.s. This holds for any $\epsilon > 0$; therefore,

$$\lim_{n\to\infty} \frac{1}{n} z_n^L = 2\gamma\mathbb{E}[c], \tag{11.56}$$

a.s., as we set out to show.

Now, by Theorem 11.3, Lemma 11.4, and (11.56), we have

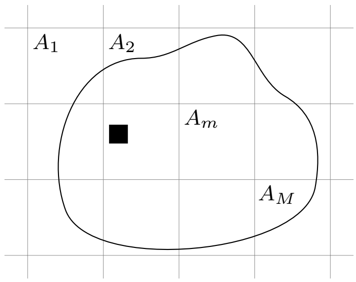
$$2\gamma\mathbb{E}[c] = \lim_{n\to\infty} \frac{1}{n} z_n^* \leq \lim_{n\to\infty} \frac{1}{n} z_n^{SC} \leq \lim_{n\to\infty} \frac{1}{n} z_n^L = 2\gamma\mathbb{E}[c].$$

Therefore,

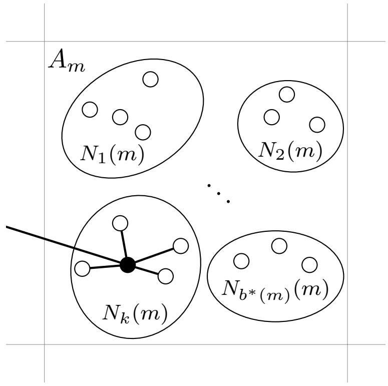
$$\lim_{n\to\infty} \frac{1}{n} z_n^{SC} = 2\gamma\mathbb{E}[c].$$

■

There is only limited computational evidence concerning the LBH's performance. Bramel and Simchi-Levi's (1995) computational results suggest that the LBH is slower and less accurate than other heuristics (see also Cordeau et al. 2002), and few, if any, other computational studies have been published. However, the heuristic has significant theoretical interest, especially in light of Theorem 11.5.

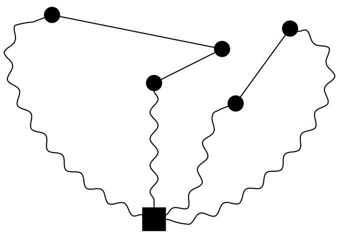


(a) Grid overlay on A.

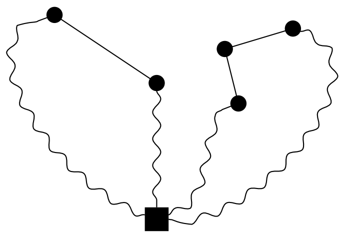


(b) Subregion A_m packed into bins.

Figure 11.10 Feasible solution for CCLP in proof of Theorem 11.5.



(a) Before exchange.



(b) After exchange.

Figure 11.11 Node exchange for VRP.

11.3.4 Improvement Heuristics

Since a solution to the VRP consists of multiple TSP-type tours, any improvement heuristic for the TSP (Section 10.5)—2-opt, Or-opt, US, etc.—can also be applied to the routes in a VRP solution.

Another important class of improvement heuristics for the VRP involves moving one or more nodes from one route to another. One simple approach searches for an individual node that can be moved to a different route to reduce the total route length (as in Figure 11.11), and repeats this process until no further exchanges can be found. We can generalize this considerably to identify multiple consecutive nodes that can be moved to another route—or exchanged with nodes on that route—to reduce the total length; see, e.g., Thompson and Psaraftis (1993) and Laporte and Semet (2001).

11.3.5 Metaheuristics

In the past few decades, a class of heuristics called *metaheuristics* has become very popular, especially for solving combinatorial optimization problems. Metaheuristics are usually quite general and can therefore apply to a wide range of problems but require customization to do so. At the core of a metaheuristic is usually one or more simpler heuristic "moves" (e.g., adding, dropping, or swapping nodes), and these simpler heuristics are manipulated by the metaheuristic itself—hence, "meta." Many metaheuristics are inspired by natural phenomena, such as the inheritance of genes, the behavior of ant colonies, the flocking of birds, and the heating and cooling of metals. These heuristics attempt to mimic nature's success in achieving certain goals by modeling their behavior algorithmically. Many incorporate randomness, just as in nature.

Classical heuristics such as insertion heuristics for the TSP (Sections 10.4.1–10.4.5), the Clarke–Wright heuristic for the VRP (Section 11.3.1), or the neighborhood search heuristic for the UFLP (Section 8.2.5) usually progress in a single direction to find a solution (though they may be randomized, seeded with different initial solutions, etc., to develop alternate solutions). In contrast, metaheuristics contain explicit mechanisms to explore more regions of the solution space, usually by allowing the search to move in directions toward inferior, or even infeasible, solutions, in the hope of then moving toward an even better solution. (Think of a mountain climber standing halfway up the shorter of two neighboring mountains; she has to go lower, first, before climbing the higher mountain.) One category of metaheuristic, called *population search*, diversifies the search by considering many solutions at a time, while another, called *local search*, does so by devoting more computational effort to improving one, or a few, solutions at a time. Because metaheuristics search harder than classical heuristics, they often produce better solutions, as well as longer computation times.

Because the VRP is so difficult to solve exactly, metaheuristics have become one of the most popular, and successful, approaches for solving them. In this section, we discuss two such methods—tabu search and genetic algorithms. For more thorough reviews of these and other metaheuristics for the VRP, including simulated and deterministic annealing, ant colony optimization, and neural networks, we refer the reader to Gendreau et al. (2001) and Vidal et al. (2013). For an overview of metaheuristics in general, see Blum and Roli (2003), Gendreau and Potvin (2010), and Luke (2013).

The success of a given metaheuristic depends heavily on a number of factors, including how it is customized for the optimization problem at hand, how it is implemented in code, and how the user sets the parameters that control its execution. This makes it difficult to compare metaheuristics to one another in general without focusing on the specific details of individual researchers' implementations. Nevertheless, tabu search is generally considered to be the most successful metaheuristic at solving VRP problems and their extensions (Gendreau et al. 2001, Cordeau and Laporte 2005). This is not to discount other metaheuristics, however—a winning simulated annealing heuristic or genetic algorithm may be just one innovation away.

11.3.5.1 Tabu Search A *tabu search* heuristic (Glover 1986, 1989, 1990), sometimes called *taboo search*, uses one or more "moves" to iterate from one solution to the next. A move is sometimes made even if it degrades the solution (or makes it infeasible), and therefore the algorithm needs a way to prevent the search from moving right back to the original solution at the next iteration. One way to do this would be simply to maintain a list of all the solutions encountered thus far and to prohibit moves that would return to one of these solutions, but this would entail an unacceptable memory and computational burden. Instead, tabu search maintains a *tabu list* of moves that are prohibited ("taboo") for a certain number of iterations. The moves on the tabu list are the reverse of the moves that were recently implemented. Many tabu search heuristics also incorporate *diversification* mechanisms to encourage new areas of the search space to be explored (e.g., by penalizing

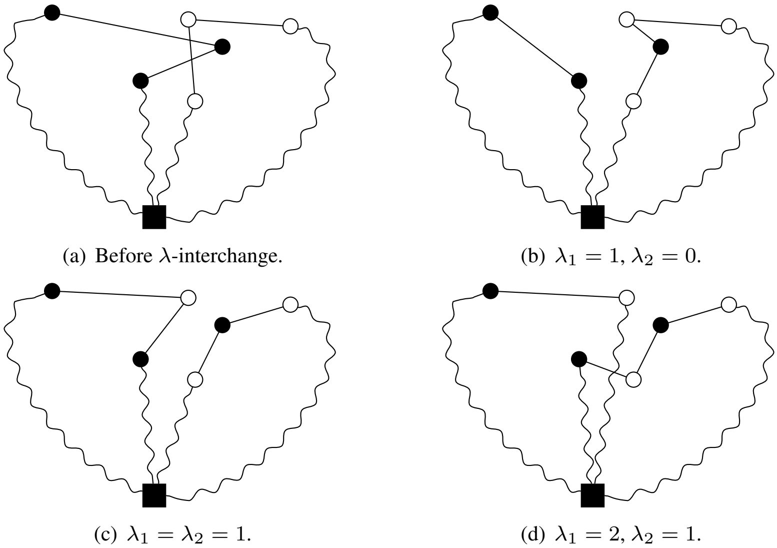


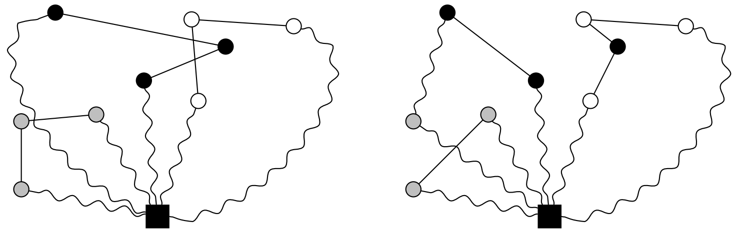
Figure 11.12 A few possible λ-interchange moves for VRP, with $\lambda = 2$. ● = node originally on route A, ○ = node originally on route B.

moves that have been made too many times already) and *intensification* mechanisms to improve promising solutions even further (e.g., by performing improvement heuristics on them). For general references on tabu search, see, e.g., Glover and Laguna (1997), and for its application to the VRP, see, e.g., Cordeau and Laporte (2005).

Perhaps the most critical decision to make when designing a tabu search heuristic is how to define the *neighborhood* of a given solution, that is, the set of solutions that can be reached from that solution via an allowable move. In the context of the VRP, a simple neighborhood might consist of all solutions that can be reached from the current solution by moving a node from its current route to a new one. The two most common moves for VRP tabu search algorithms are more flexible and powerful than this simple one.

The first is a λ*-interchange* (Osman 1993), which consists of moving λ_1 nodes from route A to route B and λ_2 nodes from route B to route A, where $\lambda_1,\ \lambda_2 \leq \lambda$ for a fixed integer λ. (See Figure 11.12.) If λ_1 or λ_2 equals 0, then we are simply moving one or more nodes from one route to another. The neighborhood of a given solution is defined as all solutions that can be reached from it via a single λ-interchange. Osman's algorithm uses $\lambda = 2$ to keep the search manageable. Taillard (1993) adds to this an intensification mechanism in which the routes are optimized using an exact TSP algorithm. His algorithm also decomposes the problem geographically so that the search can be parallelized. Rochat and Taillard (1995) enhance Taillard's (1993) algorithm using a concept that has come to be known as *adaptive memory*, and the resulting algorithm finds the best known solution for all 14 of the benchmark instances by Christofides et al. (1979).

Moves in the TABUROUTE heuristic (Gendreau et al. 1994) are very simple—they consist of moving only a single node to another route, a special case of 1-interchanges—but the heuristic compensates for the simple neighborhood structure through a host of other features. Most notably, the insertion of the node into its new route is performed using the GENI heuristic (Section 10.4.5), and intensification occurs via the US heuristic



(a) Before ejection chain. (b) After ejection chain.

Figure 11.13 Ejection chain move for VRP. ● = node originally on route A, ○ = node originally on route B, ◎ = node originally on route C.

(Section 10.5.3), both by Gendreau et al. (1992). TABUROUTE also allows infeasible solutions to be considered during the search. Solutions are evaluated based on a weighted sum of the usual VRP objective function and terms quantifying the capacity and route-length constraint violations. The weights on these terms are adjusted dynamically in the algorithm to nudge the search back toward feasibility if it has found only infeasible solutions for a certain number of iterations. Toth and Vigo (2003) propose a TABUROUTE-like algorithm that automatically eliminates long edges, since these are unlikely to appear in optimal solutions. Their approach, known as *granular tabu search*, results in shorter run times with a minor degradation in solution quality.

The second main type of move, called an *ejection chain* (Rego and Roucairol 1996, Xu and Kelly 1996), consists of moving nodes from route A to route B, other nodes from B to C, others from C to D, and so on. (See Figure 11.13.) Ejection chains are therefore generalizations of λ-interchanges. This mechanism is used in tabu search heuristics by Xu and Kelly (1996), Rego and Roucairol (1996), and Rego (1998); these heuristics appear not to perform as well as those using λ-interchanges (Cordeau and Laporte 2005).

11.3.5.2 Genetic Algorithms A *genetic algorithm* (GA; Holland 1992) is a metaheuristic in which solutions to the optimization problem are represented as genes that are passed from one generation to the next. Through a process that mimics natural selection (or survival of the fittest), good solutions are more likely to reproduce, and therefore the population tends to produce fitter and fitter offspring as it evolves.

A GA maintains a current *population* that consists of multiple *chromosomes* (or *individuals*), each of which corresponds to a solution by representing it as a string of *genes*. In each iteration of the GA, several processes (called *operators*) act on the current population to create a new one. Common operators include *reproduction*, in which good solutions from the current generation are copied to the next; *crossover*, in which information from two "parent" solutions is merged to create one or more "offspring" solutions; and *mutation*, in which a small number of genes are randomly altered. In many GAs, some or all individuals from the population are also subjected to improvement heuristics; this approach is sometimes called a *memetic algorithm*.

Let's return to the uncapacitated fixed-charge location problem (UFLP; Section 8.2) to see how a simple GA might work. Recall that a solution to the UFLP consists of variables x_j and y_{ij} that indicate whether facility j is open and whether customer i is assigned to facility j, respectively. Once we know the facility locations, the optimal assignments are easy to determine, so it suffices to encode only the x_j variables. This can be done quite

$$
\begin{array}{cccccccc}
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0
\end{array}
$$

(a) A small population.

$$
\begin{array}{ccc|ccccc}
\boxed{1} & \boxed{1} & \boxed{0} & 1 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & \boxed{0} & \boxed{1} & \boxed{1} & \boxed{0} & \boxed{1} \\
\end{array}
$$

$$\Downarrow$$

$$
\begin{array}{ccc|ccccc}
1 & 1 & 0 & 0 & 1 & 1 & 0 & 1
\end{array}
$$

(b) One-point crossover.

$$
\begin{array}{cccccccc}
1 & \boxed{1} & 0 & 1 & \boxed{0} & 0 & 1 & \boxed{0} \\
\boxed{0} & 1 & \boxed{0} & \boxed{0} & 1 & \boxed{1} & \boxed{0} & 1
\end{array}
$$

$$\Downarrow$$

$$
\begin{array}{cccccccc}
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0
\end{array}
$$

(c) Uniform crossover.

Figure 11.14 A simple genetic algorithm for the UFLP. In (b) and (c), genes selected for inheritance by the offspring are in boxes.

$$
\begin{array}{ccc|ccccc}
\boxed{6} & \boxed{2} & \boxed{1} & 3 & 7 & 4 & 5 & 8 \\
7 & 1 & 8 & \boxed{4} & \boxed{5} & \boxed{2} & \boxed{6} & \boxed{3}
\end{array}
$$

$$\Downarrow$$

$$
\begin{array}{cccccccc}
6 & 2 & 1 & 4 & 5 & 2 & 6 & 3
\end{array}
$$

Figure 11.15 One-point crossover for VRP leading to infeasible solution.

simply, by setting the gene for facility j equal to x_j. A small population for a problem with $|J| = 8$ is shown in Figure 11.14(a). Crossover can be performed in a number of ways. One way is to choose a "crossover point," and to use the genes from parent A before the crossover point and from parent B after the crossover point, as in Figure 11.14(b) (the parents are the second and fourth individuals); this is called *one-point crossover*. Another way, called *uniform crossover*, is to choose a parent randomly and independently with some probability for each gene, as in Figure 11.14(c).

Neither of these approaches works for the VRP, however. To see why, imagine we encode solutions by listing the customer nodes in the order they are visited. (For simplicity, we'll temporarily assume there is only a single route.) Then the offspring produced by simple crossover methods such as one-point or uniform crossover are likely to contain some nodes twice and some nodes not at all. Figure 11.15 illustrates this for one-point crossover. GAs for the VRP must therefore use more sophisticated crossover mechanisms to ensure feasibility.

Van Breedam (1996) proposes encoding a solution as a string in which the depot is repeated each time a new route begins. For example, the string 0 4 6 0 3 5 2 1 represents two routes, $0 \rightarrow 4 \rightarrow 6$ and $0 \rightarrow 3 \rightarrow 5 \rightarrow 2 \rightarrow 1$. His crossover mechanism is based on the partially matched crossover (PMX) operator for the TSP and other sequencing

$$
\begin{array}{cccccccccccc}
0 & 4 & \| \boxed{7} & \boxed{3} \| & 0 & 2 & 1 & 0 & 5 & 6 & 8 \\
0 & 5 & 1 & 0 & 2 & 0 \| & \boxed{6} & \boxed{8} \| & 3 & 4 & 7
\end{array}
$$

$$\Downarrow$$

$$
\begin{array}{cccccccccccc}
0 & 4 & | \; 6 & 8 \; | & 0 & 2 & 1 & 0 & 5 & 7 & 3 \\
0 & 5 & 1 & 0 & 2 & 0 \; | & 7 & 3 \; | & 8 & 4 & 6
\end{array}
$$

Figure 11.16 PMX-based crossover operator (Van Breedam 1996).

problems (Goldberg 1989). It works by selecting two crossover points and exchanging the strings between them to produce two new offspring. For example, in Figure 11.16, nodes 7 and 3 in parent A are swapped with nodes 6 and 8 in parent B, both between the crossover points and outside of them. He proposes a mutation operator in which two nodes on different routes are swapped, like a λ-interchange operation with $\lambda_1 = \lambda_2 = 1$.

A more complex, and effective, crossover operator is used in the memetic algorithm by Nagata and Bräysy (2009) and based on the edge assembly crossover (EAX) operator for the TSP (Nagata and Kobayashi 1997). Given two parents, A and B (Figure 11.17(a)), the operator works as follows:

1. Let G_{AB} be the graph consisting of node set N and edge set $(E_A \cup E_B) \setminus (E_A \cap E_B)$, where E_A and E_B are the edges in parents A and B, respectively. That is, G_{AB} contains all the edges from both parents, excluding edges that they have in common. (Figure 11.17(b).)

2. Partition G_{AB} into cycles that consist of alternating edges from parents A and B; these are called AB-cycles.

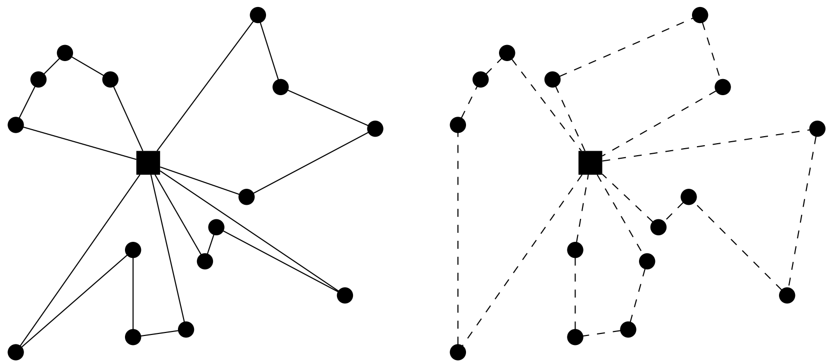
3. Choose a subset of the AB-cycles; this is called an E-set. (Figure 11.17(c).)

4. Form an intermediate solution by removing from A all edges that are in the E-set and replacing them with edges from the E-set that came from B. The intermediate solution may include subtours. (Figure 11.17(d).)

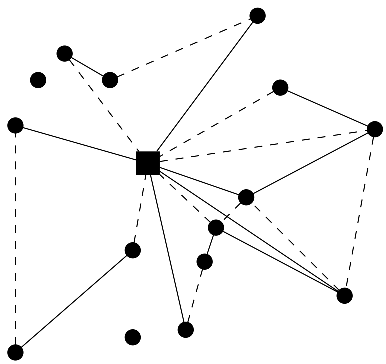
5. Fix infeasibilities with respect to subtours and capacity constraints using moves similar to 2-opt and λ-interchange.

Their method allows capacity-infeasible solutions to diversify the search space (as in a tabu search heuristic), with a penalty in the objective function to encourage feasibility. A local search procedure improves the solutions found by the GA, again using 2-opt and λ-interchange-type moves. The method found new best known solutions on several benchmark instances.

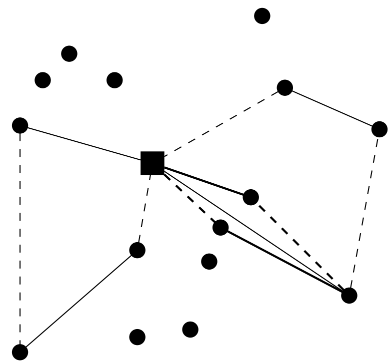
Many more varieties of GA (and the more general category of evolutionary algorithms) have been proposed for the VRP. For a thorough review, see Potvin (2009). For a general reference on GAs, see, e.g., Goldberg (1989).



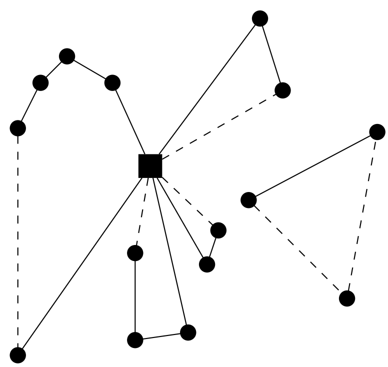
(a) Parents A and B.



(b) Graph G_{AB}.



(c) An E-set. (One cycle is drawn heavier to distinguish it from the cycle it intersects.)



(d) Intermediate solution.

Figure 11.17 EAX-based crossover operator (Nagata and Bräysy 2009).

11.4 BOUNDS AND APPROXIMATIONS FOR THE VRP

In Section 11.4.1, we discuss bounds that relate the optimal objective function value for a VRP instance to that of the corresponding TSP instance. Then, in Section 11.4.2 we discuss the asymptotic behavior of the optimal VRP objective as $n \to \infty$. Throughout this section, we assume that the number of vehicles is unrestricted.

11.4.1 TSP-Based Bounds

Suppose that each customer has a demand of 1, so that C represents the number of customers that each vehicle can serve. Let $z^*(M)$ be the total length of the optimal VRP routes through a set $M \subseteq N^-$ of customers, and let $z^T(M)$ be the length of the optimal TSP tour through the nodes in M. Let $z^T = z^T(N)$, and $z^* = z^*(N^-)$. Haimovich and Rinnooy Kan (1985) prove the following bounds on z^*:

Theorem 11.6 *For a VRP instance on nodes $N = \{0, 1, \ldots, n\}$ with $d_i = 1$ for all $i = 1, \ldots, n$,*

$$\max\left\{2\frac{n}{C}\bar{c}, z^T\right\} \le z^* \le 2\left\lceil\frac{n}{C}\right\rceil\bar{c} + \left(1 - \frac{1}{C}\right)z^T, \tag{11.57}$$

where $\bar{c} = \frac{1}{n}\sum_{i=1}^{n} c_{0i}$ is the average distance from the depot to the customer nodes.

Proof. We prove the lower bound first. Let N_k be the set of customers served by vehicle k in the optimal solution. Then

$$z^T(N_k \cup \{0\}) \ge 2\max_{i \in N_k}\{c_{0i}\} \ge 2\frac{\sum_{i \in N_k} c_{0i}}{|N_k|} \ge 2\frac{\sum_{i \in N_k} c_{0i}}{C}.$$

The first inequality follows from the triangle inequality; see Figure 11.18(a). The second inequality holds because the maximum distance from the depot to nodes in N_k is no less than the average distance, and the third follows from the capacity constraint $|N_k| \le C$. Then

$$z^* = \sum_k z^T(N_k \cup \{0\}) \ge \sum_k 2\frac{\sum_{i \in N_k} c_{0i}}{C} = \frac{2}{C}\sum_{i \in N} c_{0i} = 2\frac{n}{C}\bar{c}.$$

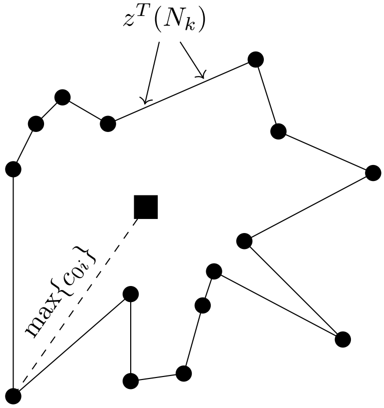
Moreover, by the triangle inequality, $z^T \le z^*$, so

$$\max\left\{2\frac{n}{C}\bar{c}, z^T\right\} \le z^*,$$

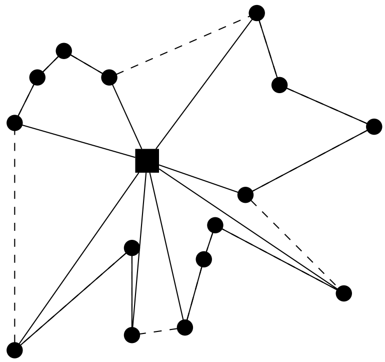
as desired.

To prove the upper bound, we describe a heuristic that constructs a feasible solution whose objective function value is at most equal to the right-hand side of (11.57). This heuristic begins with the optimal TSP tour through N^- and greedily partitions it into routes that satisfy the capacity constraint, connecting each to the depot. Each route consists of consecutive nodes from the TSP tour. Let $\ell = \left\lceil\frac{n}{C}\right\rceil$ be the resulting number of routes. This heuristic is called *optimal tour partition* (OTP).[5] (See Figure 11.18(b).) Note that ℓ edges

[5]"Optimal" refers to the optimality of the TSP tour that initializes the heuristic, not to the optimality of the partition or of the resulting VRP solution, since these may be suboptimal.



(a) $z^T(N_k \cup \{0\}) \geq 2\max_{i\in N_k}\{c_{0i}\}$.



(b) Solution produced by optimal tour partition heuristic with $n = 15$, $C = 4 \implies \ell = 4$.

Figure 11.18 Figures for proof of Theorem 11.6.

in the original TSP tour are omitted from the VRP solution; these are marked as dashed lines in Figure 11.18(b).

Now suppose that we repeat the OTP heuristic n times, each time starting at a different customer node, and then choose the best of the n solutions. We use the same orientation of the tour (clockwise or counterclockwise) each time. Call this the *iterated optimal tour partition* (IOTP) heuristic, and let z^{IOTP} be the objective value of the solution it returns; then

$$z^* \leq z^{IOTP}. \tag{11.58}$$

Each customer appears *first* on exactly ℓ routes and *last* on exactly ℓ routes, among all n solutions produced during the IOTP heuristic. Also, each edge on the TSP tour is omitted ℓ times and therefore included $n-\ell$ times among the n solutions. Therefore, the total length of all n solutions is equal to

$$2\ell \sum_{i\in N} c_{0i} + (n-\ell)z^T.$$

The total length of the best solution must be less than or equal to the average, i.e.,

$$z^{IOTP} \leq 2\ell\bar{c} + \left(1 - \frac{\ell}{n}\right) z^T. \tag{11.59}$$

Combining (11.58), (11.59), and the fact that $\ell = \left\lceil \frac{n}{C} \right\rceil$, we have the desired upper bound. ■

The upper bound in Theorem 11.6 consists of two parts; the first roughly corresponds to the "radial" distance to travel from the depot to the route, while the second represents the "local delivery" distance once the vehicle has reached the customer area.

Both bounds in Theorem 11.6 are tight—see Problem 11.18. In fact, as $C \to \infty$, the lower and upper bounds both approach z^T—that is, the VRP approaches the TSP, as we noted in Section 11.1.2.

In the special case in which the customers are uniformly distributed in the unit square and the depot is located at its center, we can approximate $z^T \approx 0.712\sqrt{n}$ using the square-root

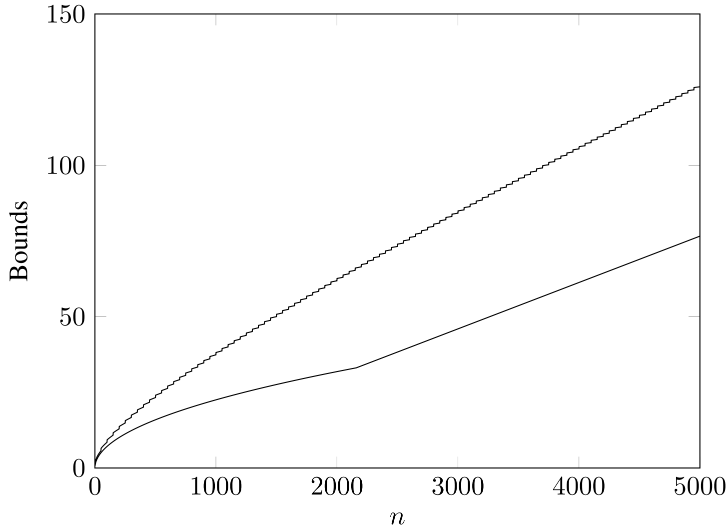


Figure 11.19 Approximate lower and upper bounds on z^* vs. n for points in unit square and $C = 50$.

approximation (10.41). One can show (Finch 2003, p. 479) that the expected distance from a random point in the unit square to the center of the square is given by

$$\frac{1}{6}\left(\sqrt{2}+\ln\left(1+\sqrt{2}\right)\right) \approx 0.383. \tag{11.60}$$

Therefore, Theorem 11.6 suggests that

$$\max\left\{0.766\frac{n}{C}, 0.712\sqrt{n}\right\} \lesssim z^* \lesssim 0.766\left\lceil\frac{n}{C}\right\rceil + 0.712\left(1-\frac{1}{C}\right)\sqrt{n} \tag{11.61}$$

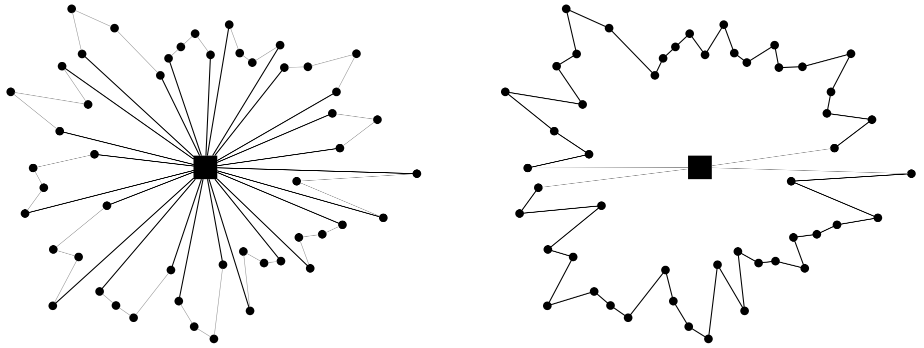
for points in the unit square. Here, the notation $a \lesssim z^* \lesssim b$ means "z^* is greater [less] than or equal to a constant that is approximately equal to a [b]." For fixed C, the approximate bounds diverge as $n \to \infty$, as shown in Figure 11.19 for $C = 50$. Note that for $n \leq 2160$, the square-root term in the lower bound dominates the $\max$, while for $n > 2160$, the linear term does.

11.4.2 Optimal Objective Function Value as a Function of n

Recall from Theorem 10.19 that as n gets large, the optimal TSP tour length increases as $\sqrt{n}$. In contrast, the optimal VRP objective function value increases linearly as n gets large. Haimovich and Rinnooy Kan (1985) prove the following:

Theorem 11.7 *Suppose that the node locations in the VRP are drawn iid from a probability distribution on a compact region with expected distance $\mathbb{E}[c]$ to the depot. Suppose that $d_i = 1$ for all $i \in N^-$. Let z_n^* be the optimal objective function value for the problem with n nodes. Then*

$$\lim_{n\to\infty} \frac{1}{n} z_n^* = \frac{2\mathbb{E}[c]}{C} \tag{11.62}$$



(a) Radial distance dominates as $n \to \infty$ without commensurate increase in C.

(b) Local delivery distance dominates as $n \to \infty$ with sufficient increase in C.

Figure 11.20 Asymptotic behavior of VRP solution as $n \to \infty$.

almost surely (a.s.).

Proof. Omitted; see Haimovich and Rinnooy Kan (1985). ■

In other words, for sufficiently large n, z_n^* can be approximated by $2\mathbb{E}[c]n/C$, the cost of using n/C vehicles and sending each to a node at a distance of $\mathbb{E}[c]$ from the depot. (We discussed a similar, but more general, result by Bramel et al. (1992) in Theorem 11.3, which allows the demands to be iid rather than equal to 1.) As we discussed in Section 11.4.1, the first term of the upper bound in Theorem 11.6 is proportional to n and represents the radial distance to travel to the customers from the depot. The second term represents the local delivery distance once we reach the set of customers on the route, and by Theorem 10.19, this term is proportional to $\sqrt{n}$ as n gets large. Theorem 11.7, then, says that for fixed C, the radial distance dwarfs the local delivery distance as $n \to \infty$, for z_n^* itself and not just its upper bound. (See Figure 11.20(a).)

What happens when C increases along with n? The answer depends on the relative rate of increase in the two parameters. Let C_n be the capacity when there are n nodes. Haimovich and Rinnooy Kan (1985) prove that, under certain conditions on the probability distributions, if C increases much more slowly than $\sqrt{n}$, then the approximate total distance for large n is still proportional to n, whereas if C increases much more quickly than $\sqrt{n}$, then the approximate total distance is proportional to $\sqrt{n}$. In other words, as $n \to \infty$, if the capacity does not keep pace with the square root of the number of nodes, then the number of routes increases and the radial distance dominates, as in Figure 11.20(a); on the other hand, if the capacity increases faster (or not much slower) than the square root of the number of nodes, then the number of routes decreases and the local delivery distance dominates, as in Figure 11.20(b). In the latter case, the solution approaches the TSP solution, which we know has a length proportional to $\sqrt{n}$ as n gets large. In fact, the constant of proportionality for the VRP distance includes the TSP constant β.

11.5 EXTENSIONS OF THE VRP

In this section, we discuss some of the more important VRP extensions that have been developed.

11.5.1 Distance-Constrained VRP

As we noted in Section 11.1.2, the VRP sometimes includes a constraint on the maximum distance or travel time for each route. In the case of travel time constraints, each node may also include a time for loading or unloading, which is included in the total travel time of the route. This variant is known as the *distance-constrained VRP* (DVRP).

The two-index formulation (11.1)–(11.6) can be adapted to handle the distance constraints by modifying the right-hand sides of the capacity-cut constraints (11.4). Calculating the best right-hand side for the revised constraints, however, is difficult because it requires solving the TSP, so an approximate value is often used. In the three-index formulation (11.10)–(11.17), we can enforce distance limits by adding constraints such as

$$\sum_{i \in N} c_{ij} x_{ijk} \le L \quad \forall k = 1, \ldots, K, \tag{11.63}$$

where L is the distance limit.

Many of the exact algorithms and heuristics we have discussed for the VRP can be easily adapted for the DVRP, as well. For example, in the Clarke–Wright savings heuristic, we can add the distance constraint to the rules for allowing route mergers.

11.5.2 VRP with Time Windows

The *VRP with time windows* (VRPTW) imposes constraints on the time that a vehicle can arrive at a given customer. For example, a manufacturing company may require that raw materials be delivered by a certain time in order to avoid running out of inventory and shutting down the production process. Or, a supermarket may require that trucks avoid peak hours so as not to clog up the parking lot during busy times. In some cases, time windows may be violated, but the shipping company incurs a penalty for doing so—these are known as *soft constraints*. Most of the literature, however, has focused on problems with *hard constraints* that cannot be violated, and we focus on the same here. See Cordeau et al. (2001) for a thorough overview of the VRPTW.

Obviously, the VRPTW is NP-hard since it is a generalization of the VRP. However, it is even NP-complete to find a *feasible* solution to the VRPTW (Savelsbergh 1985). (In contrast, it is always easy to find a feasible solution to the VRP, if one exists: We simply place each customer on its own route.)

The standard formulation for the VRPTW (see, e.g., Cordeau et al. 2001) is nonlinear; it can be linearized, but only at the expense of adding "big-M" terms, which weaken the formulation (Problem 11.13). Lower bounds can be obtained from the LP relaxation or by relaxing the time window and capacity-cut constraints, resulting in a problem that is easy to solve, but both of these bounds are typically weak. These bounds can be tightened using decomposition approaches such as Lagrangian relaxation (Kohl and Madsen 1997), column generation/set covering (Desrochers et al. 1992, Bramel and Simchi-Levi 1997), and variable splitting (also known as Lagrangian decomposition) (Kohl 1995). In principle, any of these can be embedded into a branch-and-cut algorithm to find exact optimal solutions, but the column generation approach lends itself well since, among other reasons, it tends to produce tighter bounds (Bramel and Simchi-Levi 1997).

Because it is so difficult, many of the algorithms for the VRPTW are heuristics. Solomon (1987) adapts the Clarke–Wright savings heuristic to handle time windows. Several improvement heuristics have been proposed, including some based on k-opt and Or-opt exchanges (Russell 1977, Savelsbergh 1985). Bramel and Simchi-Levi (1996) extend their (1995) location-based heuristic to the VRPTW and show that it is asymptotically optimal. In addition, a wide variety of metaheuristics have been introduced; see, e.g., Gendreau et al. (2008) for a review.

11.5.3 VRP with Backhauls

After delivering to its customers, a vehicle might then stop at additional nodes to pick up items that must be returned to the depot. For example, trucks that deliver products to individual stores in a supermarket chain may then stop at the chain's suppliers to pick up products to replenish the inventory at the depot. Doing so can save time, fuel, and mileage compared to a strategy in which the two types of customers are visited on separate sets of routes. This variant of the VRP is called the *VRP with backhauls* (VRPB). The outbound customers are called *linehaul customers*, while the return customers are called *backhaul customers*. The VRPB usually assumes that all of the linehaul customers must be served before any of the backhaul customers can be visited (because of their higher priority and because of the logistics of loading and unloading trucks).

Construction heuristics for the VRPB include methods based on the Clarke–Wright savings heuristic (Deif and Bodin 1984) and on the notion of space-filling curves (Goetschalckx and Jacobs-Blecha 1989). Toth and Vigo (1999) propose an improvement heuristic based on edge exchanges. Exact algorithms make use of set partitioning (Mingozzi et al. 1999) and Lagrangian relaxation (Toth and Vigo 1997). See Toth and Vigo (2001c) for a review of the VRPB.

11.5.4 VRP with Pickups and Deliveries

In the *VRP with pickups and deliveries* (VRPPD), each customer specifies both a pickup point and a delivery point, and the same vehicle must visit both points, in order. A typical example is a courier service. In the important case in which the "cargo" are people, the problem is known as the *dial-a-ride* problem, which is often used for transportation systems for elderly patrons who do not drive and use the service to shop, visit the doctor, and so on. For this reason, the VRPPD often also includes time window constraints. For a review, see Desaulniers et al. (2001).

11.5.5 Periodic VRP

In many practical routing problems, customers must receive deliveries with a certain frequency, but they do not specify which time periods must be used. For example, fuel delivery customers may require service once per month, supermarkets may request deliveries three times per week, and a university department may receive campus mail delivery twice per day. The problem of assigning customers to periods and to routes on those periods, and of optimizing the routes in each period, is called the *periodic VRP* (PVRP). The problem was introduced by Beltrami and Bodin (1974). For a review, see Francis et al. (2008).

Russell and Igo (1979) propose a construction heuristic based on the Clarke–Wright savings heuristic and an improvement heuristic based on the Lin–Kernighan heuristic for the TSP. Christofides and Beasley (1984) approximate the routing cost of the problem using a median problem (in a manner reminiscent of the location-based heuristic), but even that approximation is solved heuristically, by first assigning customers to periods and then building the routes. Tan and Beasley (1984) also use a two-phase approach, first assigning customers to periods and then solving a separate VRP for each period. Metaheuristics are also popular for the PVRP. For example, Cordeau et al. (1997) propose a tabu search algorithm that uses moves based on the GENI insertion heuristic, much like TABUROUTE. Their heuristic also applies to the multidepot VRP (discussed below) and a periodic version of the TSP. Drummond et al. (2001) propose a genetic algorithm coupled with a local search procedure, implemented using parallel computing to alleviate the high computational burden. Francis and Smilowitz (2006) propose a continuous approximation for a variant of the PVRP.

Only a few exact algorithms have been proposed for the PVRP. Francis et al. (2006) show that the PVRP can, under certain conditions on the periodic schedules, be simplified somewhat, and they solve the simplified model using Lagrangian relaxation. The Lagrangian subproblem decomposes into a capacitated assignment problem and a TSP variant known as the prize-collecting TSP. They are able to solve small instances of the problem to optimality. Mourgaya and Vanderbeck (2007) present an approach based on column generation in which the pricing problem must be solved heuristically.

Another variant of the VRP, called the *multidepot VRP* (MDVRP), assumes that there are multiple depots, rather than the single depot assumed in the classical VRP, and chooses which customers to serve from each depot, as well as the routing of the vehicles from each depot. Cordeau et al. (1997) show that this problem is actually a special case of the PVRP, in which the depots in the MDVRP correspond to periods in the PVRP. Any algorithm for the PVRP can therefore be used to solve the MDVRP.

CASE STUDY 11.1 ORION: Optimizing Delivery Routes at UPS

UPS is one of the largest logistics companies in the world, with annual revenues topping $50 billion, nearly two-thirds of which comes from its small-package business. Roughly 55,000 UPS drivers deliver an average of 16 million packages per day in the United States. UPS has one of the largest corporate industrial engineering (IE) departments in the world and has a long history of using operations research (OR) in its planning and operations. In 2003, the company began to develop a software package called On Road Integrated Optimization and Navigation (ORION), which today is used to optimize delivery routes for UPS's entire US network. In 2016, UPS and ORION won the INFORMS Edelman Award for operations research in practice. Holland et al. (2017) describe the project in detail.

Each morning at each of UPS's 1400 package centers in the United States, the packages to be delivered that day are assigned to delivery vans, and the customers receiving the packages are sequenced into a delivery route. Van-assignment is essentially the clustering phase of the VRP, while sequencing is the routing phase. UPS separates these two phases: Once van-assignment is complete, ORION solves a TSP-type problem to optimize the single route for each van. However, the routes are optimized every minute as the vans are being loaded, which allows planners to adjust the van assignments to rebalance the loads. Therefore, the VRP is solved as an iterative process, alternating between the clustering and routing phases.

The routing problem is solved as a TSP with time windows (TSPTW). Although the TSPTW can be formulated as a mixed-integer programming (MIP) problem, solving such a problem for the 140–160 customers on a typical route would be far too slow. In fact, none of the TSPTW algorithms in the literature, even heuristics, were fast enough for UPS's purposes. This is because the route optimization is not performed until after the van has been fully loaded, at which point the driver is ready to depart. If each driver has to wait one minute each day for the route optimization to finish, it would cost UPS an estimated $15 million per year in lost productivity. Therefore, UPS wanted to develop very fast heuristics.

After much experimentation, the heuristic finally implemented in ORION contains aspects of k-opt, Lagrangian relaxation, simulated annealing, and other metaheuristics. The algorithm implements various side constraints, such as requiring certain stops to be carried out in a strict order. It also accommodates practical preferences through penalties in the objective function. For example, the algorithm penalizes routes for zig-zagging back and forth across a busy street rather than delivering first on one side and then on another, which improves efficiency as well as driver safety. It also contains features for ensuring some consistency from one day to the next, a characteristic that is valued by both drivers and customers.

ORION was developed over roughly 10 years and cost an estimated $295 million to develop and deploy. The project involved extensive testing, both in computer experiments and on the road, and was rolled out using careful change-management strategies. Roughly 700 people worked full-time to deploy ORION, in addition to support from 100 others. UPS estimates that ORION will save an estimated $300–$400 million per year. It will result in 100 million fewer miles driven per year, corresponding to a savings of 10 million gallons of fuel and 100,000 metric tons of CO_2 emissions. ORION is run on 300 servers, and another 300 are hosted at a second data center as a backup. This infrastructure is capable of performing 30,000 route optimizations per minute. It is fitting that analytics guru Tom Davenport called ORION "arguably the world's largest operations research project" (Davenport 2013).

PROBLEMS

11.1 (VRP Construction Heuristics #1) Use each of the construction heuristics listed below to find solutions for the VRP instance shown in Figure 11.21. Coordinates and demands for the nodes in the figure are given in Table 11.2. Distances between nodes are Euclidean. The vehicle capacity is 100. For each heuristic, report the tour found and its length.

- **a)** Clarke–Wright savings heuristic
- **b)** Sweep heuristic
- **c)** Location-based heuristic

11.2 (VRP Construction Heuristics #2) Repeat Problem 11.1 for the VRP instance shown in Figure 11.22, consisting of the 20 largest cities in China. Coordinates and demands for the nodes in the figure are given in Table 11.3. The depot is in Nanjing (node

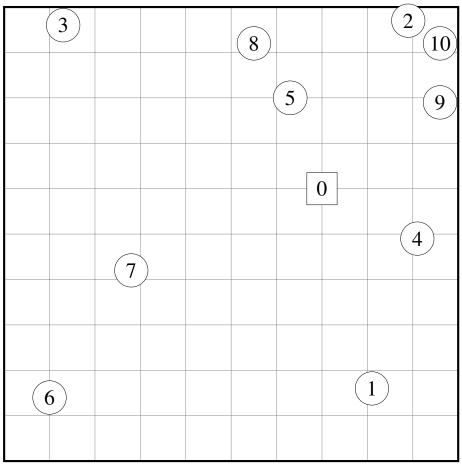


Figure 11.21 VRP instance for Problem 11.1. Distances are Euclidean.

Table 11.2 Node coordinates for Problem 11.1.

i	x_i	y_i	d_i	i	x_i	y_i	d_i
0	7.0	6.0					
1	8.1	1.6	6.3	6	1.0	1.4	16.8
2	8.9	9.7	38.8	7	2.8	4.2	36.6
3	1.3	9.6	19.4	8	5.5	9.2	21.7
4	9.1	4.9	32.0	9	9.6	7.9	8.3
5	6.3	8.0	5.7	10	9.6	9.2	38.4

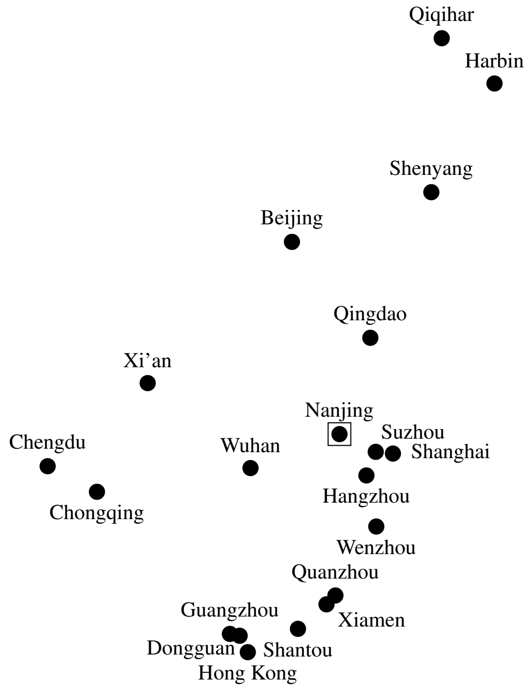


Figure 11.22 VRP instance for Problem 11.2. Distances are great circle distances.

Table 11.3 Node coordinates and demands for Problem 11.2.

i	Name	Latitude	Longitude	d_i
1	Guangzhou	23.1300	113.2590	11,071
2	Shanghai	31.2253	121.4889	22,265
3	Beijing	39.9059	116.3913	19,295
4	Shantou	23.3653	116.6949	1,347
5	Chengdu	30.6766	104.0613	10,358
6	Dongguan	23.0449	113.7525	11,090
7	Hangzhou	30.2757	120.1505	7,677
8	Wuhan	30.5960	114.2993	8,221
9	Shenyang	41.8045	123.4278	6,242
10	Xi'an	34.2192	109.1102	9,781
11	Nanjing	32.0609	118.7916	6,256
12	Hong Kong	22.2793	114.1628	6,501
13	Chongqing	29.5586	106.5493	7,166
14	Quanzhou	24.9039	118.5851	7,055
15	Wenzhou	28.0222	120.6484	7,458
16	Qingdao	36.0895	120.3497	1,398
17	Suzhou	31.2985	120.6222	3,040
18	Harbin	45.7657	126.6161	4,587
19	Qiqihar	47.3385	123.9512	5,349
20	Xiamen	24.4974	118.1356	6,705

Table 11.4 Node coordinates and demands for Problem 11.3.

i	x_i	y_i	d_i
0	1.3	1.1	—
1	4.9	1.1	500
2	3.8	0	800
3	6.2	1.2	350
4	6.4	4.6	500
5	7.8	0.9	850
6	9.3	5.2	500
7	3.8	9.6	750

Table 11.5 Partial savings list for Problem 11.4.

i	j	Savings from Merging i and j
6	7	120
3	4	105
2	13	100
1	13	90
10	11	80
8	9	75
7	8	60

11); set the demand for Nanjing to 0. Set c_{ij} equal to the great circle distance between nodes i and j (see Section 8.2.2). The vehicle capacity is 40,000.

11.3 (VRP Construction Heuristics #3) Repeat Problem 11.1 for the 8-node VRP instance whose x- and y-coordinates are given in Table 11.4. The depot is node 0. The table also lists the demand for nodes 1–7. The distance c_{ij} is the Euclidean distance between i and j. The vehicle capacity is 1500 units.

11.4 (Clarke–Wright Iterations #1) Consider the instance of the VRP pictured in Figure 11.23. The truck capacity is 300, and the demand of each customer is indicated next to it. Suppose we have performed several instances of the Clarke–Wright savings heuristic and have arrived at the solution pictured in the figure. Draw or write the routes that will result after proceeding through the portion of the savings list given Table 11.5, and indicate the total distance of the routes.

11.5 (Clarke–Wright Iterations #2) Repeat Problem 11.4 for the VRP instance shown in Figure 11.24 using the savings list in Table 11.6. The vehicle capacity is 30.

11.6 (Sweep Heuristic #1) Execute the sweep heuristic for the instance in Problem 11.4. Draw or write the resulting routes and their total distance.

11.7 (Sweep Heuristic #2) Execute the sweep heuristic for the instance in Problem 11.5. Draw or write the resulting routes and their total distance.

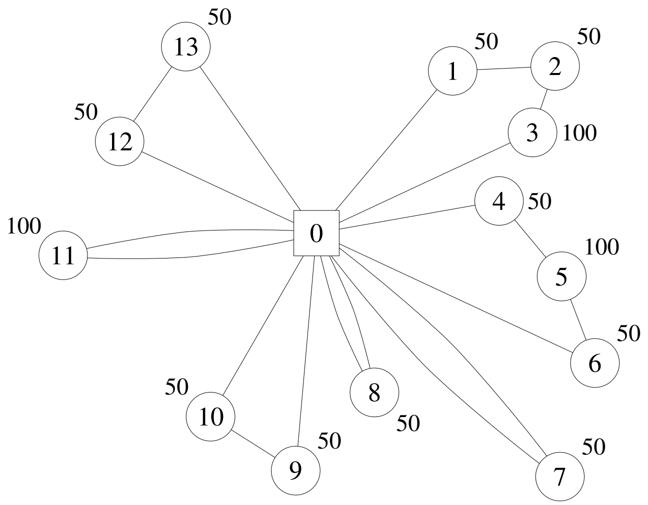


Figure 11.23 VRP instance for Problem 11.4.

Table 11.6 Partial savings list for Problem 11.5.

i	j	Savings from Merging i and j	i	j	Savings from Merging i and j
8	11	6.55	9	12	5.79
8	10	6.54	1	3	5.73
2	3	6.28	8	12	5.37
10	12	6.26	3	4	5.31
1	11	6.04	2	12	5.03
7	9	5.97	7	10	4.89
5	6	5.83	7	11	4.61

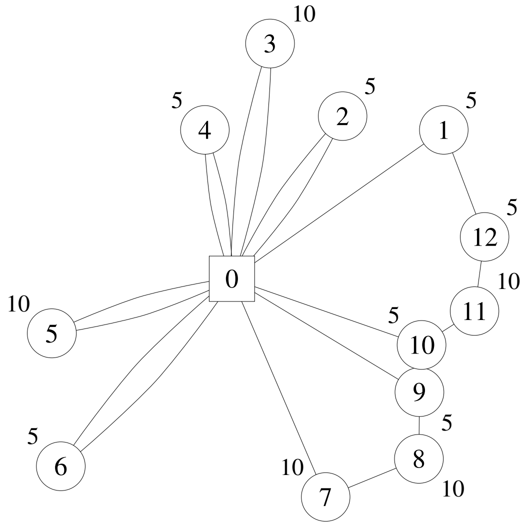


Figure 11.24 VRP instance for Problem 11.5.

11.8 (Worst-Case Error Bound for IOTP) Prove that

$$\frac{z^{IOTP}}{z^*} \le 1 + \left\lceil \frac{n}{Q} \right\rceil \frac{C-1}{n},$$

where z^{IOTP} is the objective function value of the solution returned by the IOTP heuristic described in the proof of Theorem 11.6 and z^* is the optimal VRP objective function value.

11.9 (Worst-Case Error Bound for Iterated η-Optimal Tour Partition) Suppose that we initialize the IOTP heuristic from the proof of Theorem 11.6 with a TSP tour produced by a heuristic with a fixed worst-case bound of η instead of with an optimal TSP tour. Call the resulting heuristic the iterated η-optimal tour partition (IηTP) heuristic. Prove that

$$\frac{z^{I\eta TP}}{z^*} \le \eta + \left\lceil \frac{n}{Q} \right\rceil \frac{C-\eta}{n},$$

where $z^{I\eta TP}$ is the objective function value of the solution returned by the IηTP heuristic and z^* is the optimal VRP objective function value.

11.10 (Upper Bound on Optimal VRP Length) Prove that, if the nodes in N are all contained in a rectangle with sides a and b, then the total optimal VRP route length satisfies

$$z^* \le \sqrt{2(n-1)ab} + 2(a+b).$$

11.11 (Capacity-Cut Constraints for $|S| \le 2$) Prove that the capacity-cut constraints (11.4) remain correct even if we change the "for all" part to $\forall S \subseteq N^- : |S| \ge 3$. That is, prove that for $|S| \le 2$, the constraint eliminates routes whose nodes are S and which are either capacity-infeasible or are subtours.

11.12 (Proof of (11.31)) Prove the validity of the comb inequality (11.31).

11.13 (VRPTW Formulation) In this problem, you will formulate the VRP with time windows. Suppose the vehicle serving customer $i \in N^-$ must arrive at node i in the time window $[a_i, b_i]$ for parameters a_i and b_i. Let τ_{ij} be a parameter indicating the travel time on edge (i, j). Let t_i be a new decision variable that equals the arrival time at customer $i \in N^-$.

Modify the three-index VRP formulation to account for time windows. Write one or more sets of linear constraints that set t_i appropriately and require it to be in $[a_i, b_i]$ for all $i \in N^-$. Explain your constraints in words. If you define any new notation, define it clearly.

11.14 (VRP with Precedence Constraints) Modify the three-index VRP formulation to handle *precedence constraints* that stipulate that customer i must be visited before customer j (on the same route) for all $i, j \in N^-$ for which $a_{ij} = 1$, where a_{ij} is a parameter. (Note that customer i does not need to be served *immediately* before customer j.)

11.15 (VRP with Conflicting Product Types) Consider a variant of the VRP in which there are three product types, A, B, and C. Each customer in N^- needs one of these three types; the type is given as an input. Product type A can be transported in a vehicle with product types B or C, but product types B and C cannot be mixed in the same vehicle. Modify the Clarke–Wright savings heuristic to handle this problem. Describe your heuristic in words, as well as in pseudocode.

11.16 (Garbage Pickup for Bethlehem, PA) The city of Bethlehem, PA, does not operate a centralized garbage collection service. Instead, each household hires a private garbage-collection company. A city resident claimed that centralizing the garbage collection service would result in fewer total truck miles than the current decentralized approach. Do you agree with this claim?

There are approximately 30,000 households in Bethlehem, which occupies approximately 20 square miles. Assume that each collection company solves a VRP to optimize its own routes and that a centralized collection service would do the same.

11.17 (Accuracy of ST and SC) Develop a small example (with one depot and no more than five customers) in which the star connection cost in the LBH, given by

$$f_j + \sum_{i \in N^-} \tilde{c}_{ij}$$

with f_j and $\tilde{c}_{ij}$ set using (11.45) and (11.46), respectively, more accurately estimates the optimal TSP cost through the depot and N^- than the seed tour cost, in which $\tilde{c}_{ij}$ set using (11.47). Then modify your example so that the seed tour cost is more accurate than the star connection cost.

11.18 (Theorem 11.6 Bounds are Tight) Prove that both bounds in Theorem 11.6 are tight.

11.19 (Numerical Experiment on Theorem 11.6) Generate 50 or more random VRP instances, solve them optimally, and plot their optimal objective function values, z^*, along with the upper and lower bounds from Theorem 11.6.

11.20 (Proof of Lemma 11.4) Prove Lemma 11.4.

11.21 (Bounds on the Search Space for DP) Prove that, in the minimization in (11.18), it is sufficient to consider $S' \subseteq S$ such that

$$d(S) - (k-1)C \leq d(S') \leq C.$$

In addition, prove that it is sufficient to calculate $\theta(S, k)$ for S and k such that

$$d(N^-) - (K-k)C \leq d(S) \leq kC$$

(assuming the number of vehicles is fixed to K).

11.22 (Generalized Capacity Constraints) Suppose $\mathcal{N} = (N_1, \ldots, N_T)$ is a set of disjoint subsets of N^-, $T \geq 2$, with $d(N_t) \leq C$ for all $t = 1, \ldots, T$. Suppose $H \subseteq N^-$ such that $N_t \subseteq H$ for all $t = 1, \ldots, T$. Let $b(H|\mathcal{N})$ be the optimal objective value for a bin-packing problem with bins of capacity C and $|H| - t$ items: for each $t = 1, \ldots, T$ there is an item with weight $d(N_t)$, and for each $i \in H \setminus \bigcup_{t=1}^{T} N_t$, there is an item with weight d_i. Prove that the *weak generalized capacity inequality* is valid:

$$\sum_{t=1}^{T} \sum_{\substack{i \in N_t, j \in \bar{N}_t \text{ or} \\ i \in \bar{N_T}, j \in N_t}} x_{ij} \geq 2t + 2(b(N^-|\mathcal{N}) - K).$$

CHAPTER 12

INTEGRATED SUPPLY CHAIN MODELS

12.1 INTRODUCTION

We have discussed various aspects of managing a supply chain, and most of the earlier chapters focus on one important decision in the supply chain while assuming the other decisions have already been made. For example, when we discuss inventory models, we ignore the facility location decision and its associated costs, whereas in the chapters dealing with location models, we ignore the inventory and shortage costs, as well as the demand uncertainty and the effects that reorder policies have on inventory and shipping costs. One reason for this disconnect is that the decision-maker may not possess detailed information about the future costs and other parameters in the supply chain when making facility location or network design decisions. Another reason is that the more decisions that are included in a single model, the more complex and hard to solve the model becomes. On the other hand, ignoring inventory, transportation, and other costs when making strategic decisions can lead to suboptimality. Significant cost savings can often be attained by optimizing several of the major cost drivers that can influence the performance of the supply chain.

Recall from Chapter 1 that supply chain decisions can be classified into three levels: *strategic*, *tactical*, and *operational*. Often, decisions are made at each level sequentially. For example, we might first optimize facility locations (a strategic decision) with the expectation that the facilities we choose will operate for roughly 10 years. Each month, we might make tactical inventory decisions at the facilities in anticipation of that month's demand. Finally, we might optimize vehicle delivery routes (an operational decision) every day. Under

Companion website: www.wiley.com/go/Snyder/SupplyChainTheory

this sequential approach, higher-level decisions ignore the lower-level considerations (we ignore inventory when optimizing facility locations, and we ignore routing when optimizing inventory), whereas the lower level takes the higher-level decisions as inputs (inventory models assume facility locations are fixed, and routing models assume inventory policies are fixed). In this chapter, we explore models that make decisions at multiple levels simultaneously.

When one considers whether to include decisions at multiple levels in the same optimization model, it is important to ask whether the decisions in the lower-level problems would affect the decisions at the higher level. For example, in Section 12.2, we consider a model that considers inventory costs when making facility location decisions. We will show that this model chooses different facility locations than we would obtain if we ignored inventory when locating facilities, and moreover, that the location-only solution is more expensive than the joint solution when inventory costs are factored in. This justifies the increased complexity of a joint location–inventory model and the computational burden required to solve it. On the other hand, suppose we were considering developing a model that chooses the locations of facilities simultaneously with the number of restrooms in each facility. It is unlikely that the locations chosen would be significantly different if each facility has 2 restrooms than they would if each facility has 12 restrooms. In this case, it is probably simpler—from both a modeling and a computational perspective—to optimize the facility locations first, and then optimize the number of restrooms in a separate model.

In this chapter, we discuss three types of integrated models: location–inventory, location–routing, and inventory–routing. We cover a location–inventory model thoroughly in Section 12.2, including details on mathematical formulations, solution approaches, and analytical properties. We then discuss some basic formulations and possible variations of location–routing and inventory–routing problems in Sections 12.3 and 12.4, respectively. We refer interested readers to more comprehensive surveys (e.g., Nagy and Salhi (2007), Shen (2007), Coelho et al. (2013), and Prodhon and Prins (2014)) for more details.

12.2 A LOCATION–INVENTORY MODEL

12.2.1 Introduction

We consider the design of a three-echelon supply chain consisting of one or more suppliers, distribution centers (DCs), and retailers. Each retailer places random demands to the DC that supplies it. The problem is to determine how many DCs to locate, where to locate them, which retailers to assign to each DC, how often to reorder at the DC, and what level of safety stock to maintain, so as to minimize total location, shipment, and inventory costs, while ensuring a specified level of service.

We assume that location costs are incurred when DCs are established. Line-haul transportation costs are incurred for shipments from a supplier to the DCs. Local transportation costs are incurred in moving the goods from the DCs to the retailers. Inventory costs are incurred at each DC and consist of the holding cost for the average inventory used over a period of time as well as safety stock inventory carried to protect against stockouts that might result from uncertain retailer demand. We assume that the retailers maintain only a minimal amount of inventory, which is ignored in the model below.

In an inventory system, both the cycle stock and the safety stock tend to be concave functions of the demand served. To take a simple example, suppose the demand is distributed as $N(\xi\mu, \xi\sigma^2)$ for $\xi \geq 0$. As we increase ξ, we scale both the mean and variance. If the cycle stock is set using the economic order quantity (EOQ) formula (we will see below why this is a reasonable assumption), then

$$\text{cycle stock} = \sqrt{2K\xi\mu/h}. \tag{12.1}$$

And, again for reasons to be discussed below, the safety stock is given by

$$\text{safety stock} = z_\alpha \sigma \sqrt{\xi}, \tag{12.2}$$

where α is the desired service level. The right-hand sides of both (12.1) and (12.2) are concave functions of ξ.

The upshot of this analysis is that, given the choice between many small facilities or few large facilities to serve a geographically dispersed demand, inventory costs would always favor the latter—both because of the economies of scale present in cycle stock costs and because of the risk-pooling effect with regard to safety stock costs (see Section 7.2). Of course, this doesn't mean we should always locate only a single DC. Rather, we must consider the fixed costs of the DCs and their locations (and hence transportation costs) before deciding how many DCs to open, and where. The *location model with risk pooling* (LMRP),[1] introduced by Daskin et al. (2002) and Shen et al. (2003), simultaneously optimizes all of these factors.

As we will see below, the LMRP is structured much like the uncapacitated fixed-charge location problem (UFLP), with two extra nonlinear terms in the objective function that represent the cycle- and safety-stock costs. Importantly, these costs are calculated *without including any decision variables to represent inventory decisions*. Despite its nonlinear (in fact, concave) objective function, the LMRP can be solved quite efficiently using extensions of algorithms for the UFLP.

Imagine a set of retailers, each with random demand. Some of these retailers will be converted to DCs, which will then serve the non-DC retailers (as well as their own demand). The discussion of the UFLP in Section 8.2 referred to "customers" instead of "retailers," but the terms are interchangeable—both refer to some source of demand. Also, by stating that some retailers will be "converted" into DCs, we are assuming that $I = J$ (using the notation of Section 8.2.2). We can make this assumption without loss of generality since if there is a retailer that is not a potential facility site, we can set its fixed cost to ∞, while if there is a potential facility site that is not a retailer, we can set its demand to 0.

The original motivation for the LMRP was a study of a Chicago-area blood bank system, in which inventories of blood platelets (an expensive and perishable component of donated blood) were being stored at individual hospitals, which ordered them from the blood bank's main headquarters. The hospitals were doing a poor job of managing these inventories: Some hospitals were routinely throwing away expired platelets because they had ordered too much, while others were chronically understocked and were requesting expensive emergency shipments from the blood bank. The hope was that certain hospitals could be established as distribution centers that would serve their own demand as well as those of

[1]The name "location model with risk pooling" is actually a bit misleading, since it suggests that risk pooling of safety stock is the only inventory aspect considered in the location model. On the contrary, the LMRP considers economies of scale in both cycle and safety stocks. In fact, the cycle stock costs tend to drive the results of the LMRP much more than the safety stock costs. However, we refer to the model as the LMRP here, as it is in the literature, to distinguish it from other location–inventory models.

nearby hospitals. This would allow the total amount of inventory to remain low while meeting the same service level requirements, due to the risk pooling effect. Moreover, the cost of shipments (both regular and emergency) would decrease since hospitals would now be located closer to their suppliers (Daskin et al. 2002).

The LMRP is not the only location–inventory model in the literature. For example, Barahona and Jensen (1998) consider a fixed cost to stock a given product at each DC; they solve their model using column generation. Their model is tractable but does not capture the costs of cycle and safety stock as accurately as the LMRP. Erlebacher and Meller (2000) formulate a much richer model, but it is highly nonlinear and they propose heuristics to solve it. Teo et al. (2001) consider a model that is similar to the LMRP but without transportation costs; they propose a $\sqrt{2}$-approximation algorithm for it. Nozick and Turnquist (2001a,b) linearize the inventory costs in their location–inventory models, and therefore do not capture economies of scale or risk pooling. Teo and Shu (2004) propose a location–inventory model that is similar to the LMRP except that demands are deterministic and the inventory costs are calculated using a power-of-two approximation developed by Roundy (1985). Naseraldin and Herer (2008) use a continuous approximation in which customers are located along a line segment and facilities have newsvendor-type costs; they solve their model analytically to determine the optimal number and locations of facilities on the line segment.

12.2.2 Problem Statement

Let I be the set of retailers, each of which faces normally distributed daily demands. Demands are assumed to be independent both among retailers and from day to day. The objective of the LMRP is to determine how many DCs to locate, where to locate them, and which retailers to assign to each DC to minimize the total expected location, per-unit, and inventory costs, while ensuring a specified level of service. Each DC receives product from a single supplier. Each DC is assumed to follow an (r, Q) policy to maintain its inventory, with a type-1 service level requirement. The cost of such a policy is calculated in the LMRP using the method described in Section 5.3.1.3: Q is set using the deterministic EOQ formula, and r is set using (5.21).

12.2.3 Notation

Define the following notation:

Set

I = set of retailers/potential DC sites

Parameters

Demand

μ_i = mean daily demand of retailer i

σ_i^2 = variance of daily demand at retailer i

Costs

f_j = fixed (daily) cost to open a DC at site j

K_j = fixed cost for DC j to place an order from the supplier, including fixed components of both ordering and transportation costs

c_j = per-unit cost for each item ordered by DC j from the supplier, including per-unit inbound transportation costs

d_{ij} = per-unit outbound transportation cost from DC j to retailer i
h_j = holding cost per unit per day at DC j

Other

L_j = lead time (in days) for orders placed by DC j to the supplier
α = desired fraction of DC order cycles during which no stockout occurs

Decision Variables

x_j = 1 if retailer j is selected as a DC, 0 otherwise
y_{ij} = 1 if retailer i is served by DC j, 0 otherwise

12.2.4 Objective Function

The objective function will be of the form

$$\text{minimize} \quad [\text{location cost}] + [\text{per-unit costs}] + [\text{cycle stock cost}] + [\text{safety stock cost}].$$

Location Cost: The fixed location cost is given by

$$\sum_{j \in I} f_j x_j. \tag{12.3}$$

Per-Unit Costs: The per-unit costs have two parts: the inbound cost (which includes both purchase and transportation costs from the supplier), given by

$$\sum_{j \in I} \sum_{i \in I} \mu_i c_j y_{ij}, \tag{12.4}$$

and the outbound cost, given by

$$\sum_{j \in I} \sum_{i \in I} \mu_i d_{ij} y_{ij}. \tag{12.5}$$

Inventory Costs: Consider a single DC j. We know from (5.22) that, under the expected-inventory-level (EIL) approximation, the optimal expected cost of an (r, Q) inventory policy with a type-1 service level constraint at DC j is given by

$$\sqrt{2K_j \mu h_j} + h_j z_\alpha \sigma \sqrt{L_j},$$

where μ and σ are the mean and standard deviation of the daily demand served by j. However, μ and σ are not known in advance; they depend on the decision variables. In particular:

$$\mu = \sum_{i \in I} \mu_i y_{ij}$$
$$\sigma = \sqrt{\sum_{i \in I} \sigma_i^2 y_{ij}}.$$

Therefore, the expected inventory cost at DC j is given by

$$\sqrt{2K_j h_j \sum_{i \in I} \mu_i y_{ij}} + h_j z_\alpha \sqrt{L_j \sum_{i \in I} \sigma_i^2 y_{ij}}. \tag{12.6}$$

The first term in (12.6) is the cycle stock cost, while the second is the safety stock cost. Put another way, the first term represents the economies of scale from consolidating DCs (and therefore orders), while the second represents the risk-pooling effect: As more demand is added to DC j, the amount of safety stock increases as the square root of the demand served. For both terms, adding a new retailer's demand to DC j moves the cost along the flatter part of the square-root curve, while establishing the retailer as its own DC starts it all over again at the steeper part.

Note that this concise formulation for the inventory cost would not be possible if we didn't have a closed-form expression for the optimal EOQ cost, as in (3.5), or safety stock cost, which derives from (4.24). If, to calculate the optimal EOQ cost, it was necessary to run an algorithm to find Q^* and then to plug Q^* into the cost function, then we'd need to include a variable for Q_j and somehow optimize these when we optimize all the other decision variables. This would complicate the model and algorithm considerably.

Combining (12.3)–(12.6), we get the following objective function:

$$\sum_{j\in I}\left[f_jx_j + \sum_{i\in I}\mu_i(c_j+d_{ij})y_{ij} + \sqrt{2K_jh_j\sum_{i\in I}\mu_iy_{ij}} + h_jz_\alpha\sqrt{\sum_{i\in I}L_j\sigma_i^2y_{ij}}\right]. \quad (12.7)$$

12.2.5 NLIP Formulation

The LMRP can now be formulated as a nonlinear integer program (NLIP):

$$\begin{aligned}
\text{(LMRP)} \quad \text{minimize} \quad & \sum_{j\in I}\Bigg[f_jx_j + \sum_{i\in I}\mu_i(c_j+d_{ij})y_{ij} + \sqrt{2K_jh_j\sum_{i\in I}\mu_iy_{ij}} \\
& \qquad + h_jz_\alpha\sqrt{\sum_{i\in I}L_j\sigma_i^2y_{ij}}\Bigg] && (12.8)\\
\text{subject to} \quad & \sum_{j\in I} y_{ij} = 1 \qquad \forall i\in I && (12.9)\\
& y_{ij} \le x_j \qquad \forall i\in I, \forall j\in I && (12.10)\\
& x_j \in \{0,1\} \qquad \forall j\in I && (12.11)\\
& y_{ij} \in \{0,1\} \qquad \forall i\in I, \forall j\in I && (12.12)
\end{aligned}$$

Constraints (12.9) require each retailer to be assigned to some DC, and constraints (12.10) require that DC to be open. Constraints (12.11) and (12.12) are integrality constraints. Notice that the LMRP has the exact same constraints as the UFLP—only the objective function is different. (Actually, in the UFLP, constraints (8.7) are nonnegativity constraints, not integrality constraints. But as we said in Section 8.2.2, the nonnegativity constraints in the UFLP can be replaced with integrality constraints without changing the problem.) This suggests that the Lagrangian relaxation method described in Section 8.2.3 might be adapted to solve the LMRP. That is exactly the approach that Daskin et al. (2002) take, and the approach we discuss in Section 12.2.6. We discuss two more approaches to solving the LMRP—one based on column generation and one based on conic optimization—in Sections 12.2.7 and 12.2.8, respectively. Our aim is to demonstrate that many of the fundamental tools of optimization can be brought to bear on complex supply chain optimization problems such as the LMRP.

12.2.6 Lagrangian Relaxation

As in the UFLP, we will solve the LMRP by relaxing the assignment constraints (12.9) to obtain the following Lagrangian subproblem:

$$
\begin{aligned}
&(\text{LMRP-LR}_\lambda)\\
&\text{minimize} \quad \sum_{j\in I}\left[f_j x_j + \sum_{i\in I}\mu_i(c_j+d_{ij})y_{ij} + \sqrt{2K_jh_j\sum_{i\in I}\mu_i y_{ij}}\right.\\
&\qquad\qquad \left. + h_j z_\alpha \sqrt{\sum_{i\in I} L_j\sigma_i^2 y_{ij}}\right] + \sum_{i\in I}\lambda_i\left(1-\sum_{j\in I} y_{ij}\right)\\
&\qquad = \sum_{j\in I}\left[f_j x_j + \sum_{i\in I}(\mu_i(c_j+d_{ij})-\lambda_i)y_{ij} + \sqrt{2K_jh_j\sum_{i\in I}\mu_i y_{ij}}\right.\\
&\qquad\qquad \left. + h_j z_\alpha \sqrt{\sum_{i\in I} L_j\sigma_i^2 y_{ij}}\right] + \sum_{i\in I}\lambda_i
\end{aligned} \tag{12.13}
$$

$$
\begin{aligned}
\text{subject to} \quad & y_{ij} \le x_j && \forall i \in I, \forall j \in I && (12.14)\\
& x_j \in \{0,1\} && \forall j \in I && (12.15)\\
& y_{ij} \in \{0,1\} && \forall i \in I, \forall j \in I && (12.16)
\end{aligned}
$$

12.2.6.1 Solving the Subproblem Just like the subproblem for the UFLP, we can decompose (LMRP-LR_λ) by j. Unfortunately, computing the benefit β_j is not as straightforward as it was for the UFLP because of the square-root terms. Instead, for each j, we need to solve the following problem:

$$
\begin{aligned}
(\text{P}_j) \quad \beta_j = \text{minimize} \quad & \sum_{i\in I}(\mu_i(c_j+d_{ij})-\lambda_i)y_{ij} + \sqrt{2K_jh_j\sum_{i\in I}\mu_i y_{ij}}\\
& + h_j z_\alpha \sqrt{\sum_{i\in I} L_j\sigma_i^2 y_{ij}} && (12.17)\\
\text{subject to} \quad & y_{ij} \in \{0,1\} \qquad \forall i \in I && (12.18)
\end{aligned}
$$

Although (P_j) is a concave integer minimization problem, it can be solved relatively efficiently—in order $O(|I|^2 \log |I|)$ time, using an algorithm developed by Shu et al. (2005). We won't discuss this algorithm. Instead, we'll discuss an even more efficient algorithm that relies on the following assumption:

Assumption 12.1 *The ratio of the demand variance to the demand mean is identical for all retailers. That is, for all $i \in I$, $\sigma_i^2/\mu_i = \gamma$ for some constant $\gamma \ge 0$.*

At first glance, this seems like an unreasonable assumption to make. However, if the demands come from a Poisson process (as is commonly assumed), Assumption 12.1 holds exactly, since the variance of the Poisson distribution equals its mean (hence $\sigma_i^2/\mu_i = 1$ for all i). Now (12.17) can be rewritten as follows:

$$\begin{aligned}
&\sum_{i\in I}(\mu_i(c_j+d_{ij})-\lambda_i)y_{ij} + \sqrt{2K_jh_j\sum_{i\in I}\mu_i y_{ij}} + h_j z_\alpha\sqrt{\sum_{i\in I}L_j\sigma_i^2 y_{ij}} \\
=&\sum_{i\in I}(\mu_i(c_j+d_{ij})-\lambda_i)y_{ij} + \sqrt{2K_jh_j\sum_{i\in I}\mu_i y_{ij}} + h_j z_\alpha\sqrt{\sum_{i\in I}L_j\gamma\mu_i y_{ij}} \\
=&\sum_{i\in I}(\mu_i(c_j+d_{ij})-\lambda_i)y_{ij} + \left(\sqrt{2K_jh_j} + h_j z_\alpha\sqrt{L_j\gamma}\right)\sqrt{\sum_{i\in I}\mu_i y_{ij}}
\end{aligned}$$

We have gotten rid of one of the square-root terms, which allows us to rewrite (P_j) as:

$$(\mathrm{P}_j') \quad \beta_j = \text{minimize} \quad \sum_{i\in I} a_i y_i + \sqrt{\sum_{i\in I} b_i y_i} \tag{12.19}$$

$$\text{subject to} \quad y_i \in \{0,1\} \qquad \forall i \in I \tag{12.20}$$

where

$$\begin{aligned}
a_i &= \mu_i(c_j+d_{ij}) - \lambda_i \\
b_i &= \mu_i\left(\sqrt{2K_jh_j} + h_j z_\alpha\sqrt{L_j\gamma}\right)^2 \\
y_i &= y_{ij}.
\end{aligned}$$

It turns out that (P_j') can be solved even more efficiently than (P_j), in $O(|I|\log|I|)$ time. We will describe the algorithm shortly. First, let $I^- = \{i \in I | a_i < 0\}$; that is, I^- is the set of retailers that have negative a_i. Let $I_1^- = \{i \in I^- | b_i > 0\}$ and $I_2^- = \{i \in I^- | b_i = 0\}$. Note that $I_1^- \cup I_2^- = I^-$ since $b_i \geq 0$ for all i. We will further assume that the elements of I_1^- are indexed and sorted such that

$$\frac{a_1}{b_1} \leq \frac{a_2}{b_2} \leq \cdots \leq \frac{a_m}{b_m},$$

where $m = |I_1^-|$. The algorithm for solving (P_j') relies on the following theorem:

Theorem 12.1 *There exists an optimal solution y^* to (P_j') such that the following property holds:*

(1) $y_i^* = 0$ for all $i \in I \setminus I^-$

Moreover, for every optimal solution y^* to (P_j'), the following two properties hold:

(2) $y_i^* = 1$ for all $i \in I_2^-$

(3) If $y_k^* = 1$ for some $k \in I_1^-$, then $y_l^* = 1$ for all $l \in \{1, \ldots, k-1\}$

Proof. (1) follows from the fact that for all i, $b_i \geq 0$; if $a_i \geq 0$ as well, then the objective function does not increase when $y_i = 1$ as opposed to $y_i = 0$. (2) follows from the fact that if $b_i = 0$ and $a_i < 0$, then setting $y_i = 1$ decreases the objective function.

To prove (3), suppose, for a contradiction, that y^* is an optimal solution such that $y^*_k = 1$ for some $k \in I_1^-$ but $y^*_l = 0$ for some $l \in \{1, \dots, k-1\}$. Define two new solutions y' and y'' as follows:

$$y'_i = \begin{cases} 1, & \text{if } i = l \\ y^*_i, & \text{otherwise} \end{cases}$$

$$y''_i = \begin{cases} 0, & \text{if } i = k \\ y^*_i, & \text{otherwise} \end{cases}$$

In other words, $y' = y^*$ except that y^*_l is changed to 1, and $y'' = y^*$ except that y^*_k is changed to 0. (See Figure 12.1.) Let z^*, z', and z'' be the objective values of y^*, y', and y'', respectively.

Let $R = \{i \in I_1^- | y^*_i = 1\}$ (see Figure 12.1), and let

$$B = \sum_{i \in R} b_i.$$

Then

$$z' - z^* = a_l + \sqrt{B + b_l} - \sqrt{B}$$

and

$$z^* - z'' = a_k + \sqrt{B} - \sqrt{B - b_k}.$$

Next, note that

$$\frac{a_l}{b_l} \le \frac{a_k}{b_k} \tag{12.21}$$

by assumption and that

$$\frac{\sqrt{B + b_l} - \sqrt{B}}{b_l} < \frac{\sqrt{B} - \sqrt{B - b_k}}{b_k} \tag{12.22}$$

by the strict concavity of the square-root function. Therefore,

$$\begin{aligned} \frac{z' - z^*}{b_l} &= \frac{a_l}{b_l} + \frac{\sqrt{B + b_l} - \sqrt{B}}{b_l} \\ &\le \frac{a_k}{b_k} + \frac{\sqrt{B + b_l} - \sqrt{B}}{b_l} \quad \text{(by (12.21))} \\ &< \frac{a_k}{b_k} + \frac{\sqrt{B} - \sqrt{B - b_k}}{b_k} \quad \text{(by (12.22))} \\ &= \frac{z^* - z''}{b_k} \\ &\le 0 \quad \text{(since } y^* \text{ is optimal and } b_k > 0). \end{aligned}$$

Since $b_l > 0$, we have that $z' - z^* < 0$. In other words, y' is a strictly better solution than y^*, violating the assumption that y^* is optimal. Therefore, y^* satisfies (3). ■

The upshot of Theorem 12.1 is that an optimal solution to (P_j) exists that has the form shown in Figure 12.2. (P_j) can therefore be solved using Algorithm 12.1. Note that, in line 5 of the algorithm, we take the sums to equal 0 if $r = 0$. The step with the most

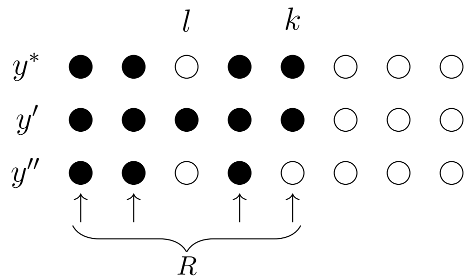


Figure 12.1 Proof of Theorem 12.1(c). Filled circles represent $y_i = 1$; open circles represent $y_i = 0$.

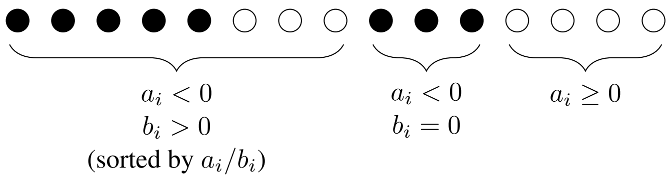


Figure 12.2 Solution to problem (P_j). Filled circles represent $y_i = 1$; open circles represent $y_i = 0$.

iterations is the sorting step in line 3, which can be performed in $O(|I| \log |I|)$ time. The algorithm, therefore, can be performed in $O(|I| \log |I|)$ time for each j. At each iteration of the Lagrangian procedure, we must solve (P_j) for each j, so the total effort required at each iteration is $O(|I|^2 \log |I|)$. To solve (LMRP-LR$_\lambda$), we set $x_j = 1$ if $\beta_j + f_j < 0$ and set $y_{ij} = 1$ if $x_j = 1$ and if $y_i = 1$ in the optimal solution to (P'_j).

Algorithm 12.1 Solve (P'_j)

1: $y_i \leftarrow 0 \ \forall i \in I \setminus I^-$ $\quad\triangleright$ Set y_i for "easy" cases
2: $y_i \leftarrow 1 \ \forall i \in I_2^-$
3: sort elements in I_1^- in increasing order of a_i/b_i $\quad\triangleright$ Calculate partial sums
4: **for all** $r \in \{0\} \cup I_1^-$ **do**
5:
$$S_r \leftarrow \sum_{i=1}^{r} a_i + \sqrt{\sum_{i=1}^{r} b_i} \tag{12.23}$$
6: **end for**
7: $r^* \leftarrow \operatorname{argmin}_r \{S_r\}$ $\quad\triangleright$ Calculate benefit
8: $y_i \leftarrow 1 \ \forall r = 1, \ldots, r^*$
9:
$$\beta_j \leftarrow \sum_{i \in I} a_i y_i + \sqrt{\sum_{i \in I} b_i y_i}$$

The proof of Theorem 12.1 did not use any special properties of the square-root function except its concavity. Therefore, Theorem 12.1 still holds if we replace the square-root term in (12.19) with *any* concave function of the total mean demand served by DC j. (Note that this square-root term is a concave function not only of $\sum_i b_i y_i$, but also of $\sum_i \mu_i y_i$, the total mean demand served by DC j, since b_i equals μ_i times a constant that is independent of i.) This means that the Lagrangian relaxation algorithm discussed here can be used to solve (LMRP) if the sum of the square-root terms in the objective function is replaced by any concave function of the total mean demand served by DC j. This property has allowed the LMRP to be extended in a number of ways; see, e.g., Qi et al. (2010).

12.2.6.2 Finding Upper Bounds As in the Lagrangian relaxation algorithm for the UFLP, at each iteration we want to convert a solution to the Lagrangian subproblem into a feasible solution for the original problem. As before, we start by opening the facilities that are open in the optimal solution to (LMRP-LR$_\lambda$). Unlike in the UFLP, however, retailers are not always assigned to their nearest open facilities in the LMRP since assignment costs are based on inventory as well as transportation. In other words, the savings from the risk-pooling effect and economies of scale may outweigh the increased transportation cost if a retailer is assigned to a more distant facility. (See Problem 12.5.) In fact, it is possible for the following strange thing to happen in an optimal solution to the LMRP: A DC is opened in, say, Chicago, but the retailer in Chicago is served by a DC in Minneapolis instead of the DC in Chicago. This would probably never happen in practice, though, so it's a little inconvenient that our model would allow this to be optimal. Fortunately, if Assumption 12.1 holds, it can be shown that this situation is never optimal.

Once we choose which DCs to open, we assign retailers to DCs as follows. First, we loop through all retailers with $\sum_j y_{ij} \geq 1$ in the optimal solution to (LMRP-LR$_\lambda$) (retailers that are assigned to at least one facility) and assign each retailer to the facility j with $y_{ij} = 1$ that minimizes the increase in cost based on the assignments already made. Next, we loop through all retailers with $\sum_j y_{ij} = 0$ and assign these retailers to the open facility that minimizes the increase in cost based on the assignments already made. Note that we only allow retailers with $\sum_j y_{ij} \geq 1$ to be assigned to DCs for which $y_{ij} = 1$, while we allow retailers with $\sum_j y_{ij} = 0$ to be assigned to any open DC.

After assigning retailers to DCs in this manner, we may want to apply two *improvement heuristics*:

- *Retailer reassignment*: For each retailer i, determine whether there is a DC that i can be assigned to instead of its current DC that would decrease the objective value. If so, reassign the retailer. If the reassignment means that the old DC no longer has any retailers assigned to it, it can be closed, saving the fixed cost, as well.

- *DC exchange*: Loop through the DCs, looking for an open DC j and a closed DC k such that if j were closed and k were opened (and retailers reassigned as needed), the objective value would decrease. If such a pair can be found, the DCs are exchanged and the heuristic continues.

12.2.6.3 Other Aspects of the Algorithm The remaining aspects of the Lagrangian relaxation algorithm (subgradient optimization, branch-and-bound) are identical to the algorithm described for the UFLP in Section 8.2.3.

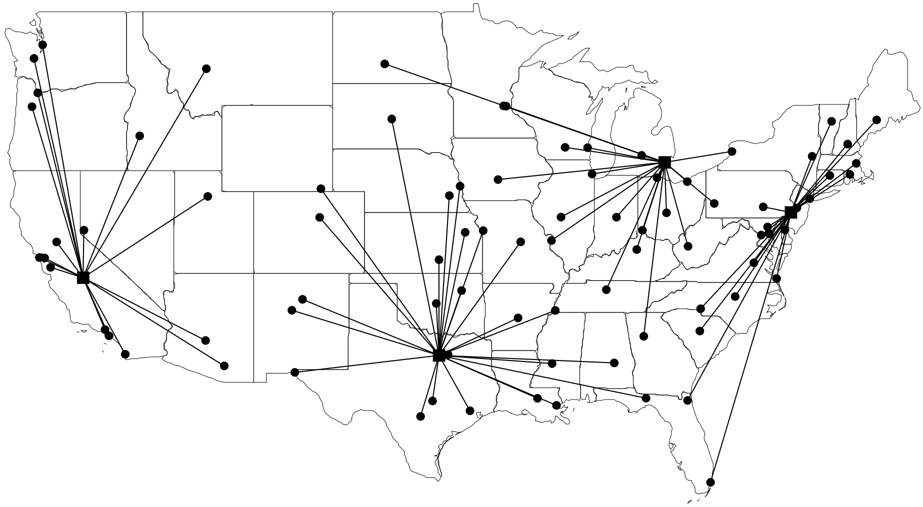


Figure 12.3 Optimal solution to 88-node LMRP instance. Total cost = $1,057,006.

Table 12.1 Costs for optimal and UFLP-based solution to 88-node LMRP instance.

Cost Type	Optimal LMRP Solution ($)	UFLP Solution ($)	% Difference
Fixed	215,200	262,100	21.8
Transportation	573,275	542,538	−5.4
Cycle stock	200,475	217,806	8.7
Safety stock	68,056	73,940	8.7
Total	1,057,006	1,096,383	3.7

☐ **EXAMPLE 12.1**

Return to the 88-node UFLP instance discussed in Example 8.1. Set the demand means μ_i equal to the demands from Example 8.1 and the variances equal to the means (that is, $\gamma = 1$). Let $K_j = 7500$, $h_j = 150$, and $L_j = 3$ for all $j \in I$. (Given the magnitude of the demands and costs in Example 8.1—i.e., demands in the hundreds and transportation costs in the \$100s or \$1000s per unit—it makes sense to assume the product units are large, like cases, and the time unit is 1 month. The parameters given above are consistent with that interpretation.) We'll set $c_j = 0$ for all j for similar reasons as we did for the EOQ in Section 3.2.2—if the purchase cost is the same at all facilities, it does not affect the optimization, but it does inflate the objective function and water down the difference between the LMRP and the UFLP. Let $\alpha = 0.975$, so $z_\alpha = 1.96$. (The full data set is available in the file `88node-lmrp.xlsx` on the book's companion web site.)

The optimal solution for the LMRP for this instance is shown in Figure 12.3. The solution opens four DCs—one fewer than the optimal UFLP solution in Example 8.1—located in Philadelphia, PA; Detroit, MI; Fort Worth, TX; and Fresno, CA. The solution has a total expected monthly cost of \$1,057,006. This cost is broken down by type in the first two columns of Table 12.1.

□

12.2.6.4 Computational Results Most papers that introduce Lagrangian relaxation algorithms test the algorithm on one or more data sets. These data sets may come from real-life problems, but more commonly, they are randomly generated since real-life data are hard to come by. There are several performance measures of interest when evaluating a Lagrangian relaxation algorithm. For example:

- How quickly does the algorithm solve the test problems? CPU times of under a minute are generally considered to be quite fast, but times of an hour or longer can be acceptable as well, depending on the context. Some people argue that for strategic problems like facility location, which might be solved only once every few years, it's acceptable for the algorithm to take several hours. Others argue that these models are typically run many times during the process of fine-tuning the data and running what-if scenarios, in which case long run times may be unacceptable.

- How large are the test problems? CPU times will be dependent on the size of the test data sets, so they should be evaluated with this in mind. Ideally, the data sets tested should include a range of sizes (in this case, number of retailers) so that the reader gets a sense of how fast the CPU time grows with the problem size and how large a problem the algorithm can handle before it gets too slow.

- How tight are the bounds achieved by the Lagrangian process, before branch-and-bound begins (i.e., at the root node of the branch-and-bound tree)? Just like in the standard LP-based branch-and-bound algorithm, it is important to have tight bounds at the root node, otherwise too much branching may be required before the optimality gap is closed.

- How many branch-and-bound nodes are required before the optimal solution is found (and proven optimal)? This goes hand-in-hand with the previous question, since large root-node gaps will probably mean that many branch-and-bound nodes are required. Note that the optimal solution may be found quite early, but many branch-and-bound nodes may be required to *prove* optimality. That is, if the root-node gap is large, it's possible that we've already found the optimal solution but that branch-and-bound will be required to prove that it is optimal.

The algorithm discussed in this section turns out to be quite efficient by these measures. Daskin et al. (2002) report that they solved problems with up to 150 nodes in under 20 seconds on a desktop computer, with no more than 3 branch-and-bound nodes required for each problem. Root-node gaps are generally less than 1%.

The authors report the following managerial insights. First, the optimal number of DCs increases as the transportation cost increases, and it decreases as the holding cost increases. (This result is not surprising, but it is important for validating the model.) Second, although the optimal solution may involve a few retailers that are assigned to facilities other than the closest (see page 521), forcing retailers to be assigned to their closest facilities (for reasons of convenience) does not generally increase the cost by too much. Finally, fewer DCs are located when inventory is taken into account. That is, a firm that solves the UFLP instead of the LMRP, ignoring inventory, will build too many DCs, because it ignores the tendency toward consolidation brought about by the economies of scale in inventory costs.

☐ **EXAMPLE 12.2**

Suppose we solve the 88-node LMRP example described in Example 12.1 as a UFLP—that is, ignoring the inventory costs. From Example 8.1, we know that the optimal UFLP solution to this instance has five DCs. Assigning retailers as described in Section 12.2.6.2 and applying the LMRP cost function (12.7), we get a total cost of \$1,096,383—3.7% greater than the optimal LMRP cost (Figure 12.4). See Table 12.1 for a breakdown of this cost by type. The UFLP-based solution has greater fixed costs (since it opens more DCs) and smaller transportation costs (since the DCs are closer, on average, to the retailers). It also has greater cycle and safety stock costs, since it uses more inventory locations. This confirms that ignoring inventory when making facility location decisions can be costly, since the resulting solution fails to achieve the economies of scale and risk pooling that a more consolidated solution could achieve.

The UFLP-based solution includes eight retailers that are assigned to DCs that are not their closest open DC. One example is Minneapolis, MN, which is 213.9 miles from the DC in Topeka, KS, but is instead assigned to the DC in Detroit, MI, which is 268.4 miles away. Why? Suppose we assigned Minneapolis to Topeka, instead. Doing so would save $\$0.50 \cdot (268.4 - 213.9) \cdot 36.8 = \$1002.8$ in transportation costs (since $h_i = 36.8$ for this retailer and c_{ij} equals half the distance between i and j). On the other hand, excluding the Minneapolis demand, the Detroit DC has 949.1 units of demand assigned, and the Topeka DC has 242.3. Reassigning Minneapolis from Detroit to Topeka would cost an extra

$$\sqrt{2 \cdot 7500 \cdot 150}\left[\left(\sqrt{242.3 + 36.8} - \sqrt{242.3}\right) + \left(\sqrt{949.1 - 36.8} - \sqrt{949.1}\right)\right] = \$805.7$$

in cycle stock costs and an extra

$$150 \cdot 1.96\sqrt{3}\left[\left(\sqrt{242.3 + 36.8} - \sqrt{242.3}\right) + \left(\sqrt{949.1 - 36.8} - \sqrt{949.1}\right)\right] = \$273.5$$

in safety stock costs. Therefore, the extra inventory costs more than offset the savings in transportation costs.

The push for consolidation can even be strong enough to abandon some of the DCs altogether. For example, if $K_j = 10{,}000$ instead of 7,500, the UFLP-based solution consolidates all of the demand into four of the five open DCs, leaving the Topeka DC unused even though its fixed cost is already accounted for.

☐

In the next two sections, we discuss two additional methods for solving the LMRP, using column generation and conic optimization.

12.2.7 Column Generation

Shen et al. (2003) present a different algorithm for solving the LMRP. Their method involves formulating the problem as a *set covering problem* and solving it using *column generation*. Here's an overview of how it works. We also refer interested readers to a brief tutorial on the technique of column generation in Appendix D.2.

First suppose that we could write down every possible subset $R \subseteq I$. (There are $2^{|I|}$ such subsets.) Let $\mathcal{R}$ be the collection of all of these subsets. For each subset $R \in \mathcal{R}$ and

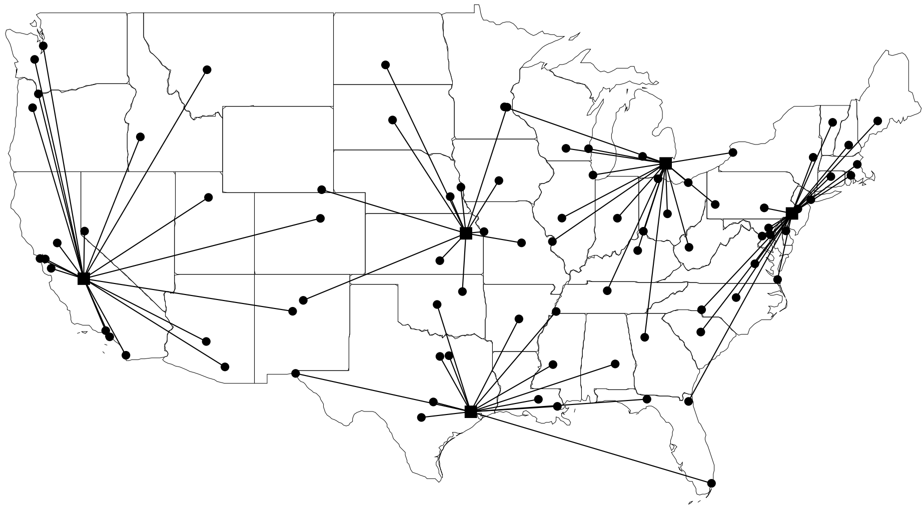


Figure 12.4 UFLP-based solution to 88-node LMRP instance. Total cost = \$1,096,383.

for each facility $j \in R$, let $c_{R,j}$ be the cost of serving all of the retailers in R from a DC located at j:

$$c_{R,j} = f_j + \sum_{i\in R} \mu_i(c_j + d_{ij}) + \sqrt{2K_j h_j \sum_{i\in R} \mu_i} + h_j z_\alpha \sqrt{\sum_{i\in R} L_j \sigma_i^2}.$$

Then choose the cheapest facility in R and call its cost c_R:

$$c_R = \min_{j\in R}\{c_{R,j}\}.$$

(Note that c_R is different from c_j: c_R is the minimum cost of serving all retailers in R, whereas c_j is simply the per-unit ordering cost at DC j.) The idea behind modeling this problem as a set covering problem is to choose several sets from $\mathcal{R}$ so that every retailer is contained in exactly one set. Each set corresponds to a group of retailers that will be served by a single facility; c_R represents the cost of this group.

The set covering model has a single decision variable for each $R \in \mathcal{R}$:

z_R = 1 if set R is in the solution, 0 otherwise

The set covering formulation of the LMRP is as follows:

$$\text{(LMRP-SC)}\quad \text{minimize} \quad \sum_{R\in\mathcal{R}} c_R z_R \tag{12.24}$$

$$\text{subject to} \quad \sum_{R\in\mathcal{R}: i\in R} z_R \ge 1 \quad \forall i \in I \tag{12.25}$$

$$z_R \in \{0,1\} \qquad \forall R \in \mathcal{R} \tag{12.26}$$

The objective function (12.24) computes the cost of all of the sets chosen. Constraints (12.25) say that every retailer must be included in at least one chosen set—that is, every retailer must be assigned to some open facility. Although (12.25) is written with a $\ge$, any optimal solution will have each retailer assigned to *exactly* one facility (why?), so the constraints could be written with an $=$. This is called a "set covering" problem because the idea is to choose a number of sets to "cover" every element in I.

It seems we have lost two aspects of the original problem in formulating it as a set covering problem. First, (LMRP-SC) is linear, while (LMRP) is nonlinear. What happened to the nonlinearity? Computing each cost c_R requires solving a nonlinear problem, so the nonlinearity in (LMRP) is present in the setup to (LMRP-SC), not in (LMRP-SC) itself. Second, nothing in the formulation of (LMRP-SC) indicates which j are chosen—only which sets are chosen. That is, if $R = \{2, 4, 7, 11\}$ and $z_R = 1$, we know that retailers 2, 4, 7, and 11 are served by the same DC, and that that DC is either 2, 4, 7, or 11, but which one is it? The answer to this question, too, is hidden in the computation of c_R. To compute c_R, we had to compute $c_{R,j}$ for $j = 2, 4, 7, 11$; whichever was smallest became c_R. Somewhere we would have recorded which j gave the best cost, and we'd use that to convert a solution to (LMRP-SC) into a solution to the original problem.

In principle, we could solve the LMRP by enumerating all the sets in $\mathcal{R}$ and then solving (LMRP-SC). Unfortunately, there are two problems with this approach. First, there are $2^{|I|}$ elements in $\mathcal{R}$—far too many to enumerate. Second, even if we could enumerate $\mathcal{R}$, it's not clear how we would solve (LMRP-SC). The solution to the first problem is to enumerate a small handful of elements of $\mathcal{R}$ first, then identify new elements as needed as the algorithm proceeds. (How do we do this? We'll find out below.) The solution to the second problem is to solve the LP relaxation of (LMRP-SC), then use branch-and-bound if the resulting solution is not integer. It turns out that the set covering problem usually has a very tight LP bound. Sometimes the solution to the LP relaxation is naturally all-integer; if not, it doesn't usually take much branching to find an optimal integer solution. So in what follows, we'll focus on solving the LP relaxation of (LMRP-SC), not (LMRP-SC) itself.

Suppose we have enumerated a subset of $\mathcal{R}$—call it $\mathcal{R}'$. We might do this by generating random sets, or using some heuristic. We need to solve the LP relaxation of (LMRP-SC) including only the sets in $\mathcal{R}'$, not all of $\mathcal{R}$. This problem is called the *restricted master problem*; we will denote it by ($\overline{\text{LMRP-SC}}$):

$$(\overline{\text{LMRP-SC}}) \quad \text{minimize} \quad \sum_{R \in \mathcal{R}'} c_R z_R \tag{12.27}$$

$$\text{subject to} \quad \sum_{R \in \mathcal{R}' : i \in R} z_R \geq 1 \quad \forall i \in I \tag{12.28}$$

$$0 \leq z_R \leq 1 \qquad \forall R \in \mathcal{R}' \tag{12.29}$$

Now suppose we solve ($\overline{\text{LMRP-SC}}$). Let $\bar{z}_R$ ($R \in \mathcal{R}'$) be an optimal solution. Recall from basic LP theory that any optimal solution to an LP has a corresponding optimal *dual solution*. Let $\bar{\pi}_i$ ($i \in I$) be the optimal dual solution corresponding to $\bar{z}$. Recall also that a solution to a minimization LP is optimal if every variable has nonnegative reduced cost with respect to the dual variables. Therefore, if we were to solve ($\overline{\text{LMRP-SC}}$) with all of $\mathcal{R}$ instead of just $\mathcal{R}'$, $\bar{z}$ would still be optimal provided that

$$c_R - \sum_{i \in R} \bar{\pi}_i \geq 0 \tag{12.30}$$

for each $R \in \mathcal{R}$. So even if we didn't solve ($\overline{\text{LMRP-SC}}$) over all of $\mathcal{R}$, we can check whether a given solution is optimal by checking (12.30) for all $R \in \mathcal{R}$. Of course, we don't want to check (12.30) for all $R \in \mathcal{R}$ since we can't enumerate $\mathcal{R}$. Instead, we search for an $R \in \mathcal{R}$ that *violates* (12.30). But how?

Let R_j^* be the set in $\mathcal{R}$ that uses j as its designated DC and has the minimum reduced cost. If R_j^* has nonnegative reduced cost for all $j \in I$, then every $R \in \mathcal{R}$ has nonnegative reduced cost. For each j, we can find R_j^* by solving the following *pricing problem*:

$$\begin{aligned}
\text{minimize} \quad & f_j + \sum_{i \in I} \mu_i (c_j + d_{ij}) y_{ij} + \sqrt{2K_j h_j \sum_{i \in I} \mu_i y_{ij}} \\
& + h_j z_\alpha \sqrt{\sum_{i \in I} L_j \sigma_i^2 y_{ij}} - \sum_{i \in I} \bar{\pi}_i y_{ij} \\
= \; & f_j + \sum_{i \in I} (\mu_i (c_j + d_{ij}) - \bar{\pi}_i) y_{ij} + \sqrt{2K_j h_j \sum_{i \in I} \mu_i y_{ij}} \\
& + h_j z_\alpha \sqrt{\sum_{i \in I} L_j \sigma_i^2 y_{ij}} && (12.31) \\
\text{subject to} \quad & y_{ij} \in \{0, 1\} \quad \forall i \in I && (12.32)
\end{aligned}$$

A solution to this problem can be converted to a set R_j^* by setting

$$R_j^* = \{i \in I | y_{ij} = 1\};$$

this set has cost

$$c_{R_j^*} = f_j + \sum_{i \in I} (\mu_i (c_j + d_{ij})) y_{ij} + \sqrt{2K_j h_j \sum_{i \in I} \mu_i y_{ij}} + h_j z_\alpha \sqrt{\sum_{i \in I} L_j \sigma_i^2 y_{ij}}.$$

Does this problem look familiar? Of course it does—this is the same problem as (P_j) (see page 517), plus a constant (f_j) and with λ_i replaced by $\bar{\pi}_i$. We already know how to solve this problem.

So, at each iteration of the algorithm, we solve ($\overline{\text{LMRP-SC}}$), then solve the pricing problem above for each j. If the objective function is nonnegative for every j, we have found the optimal solution to ($\overline{\text{LMRP-SC}}$). If it is negative for some j, then we add the corresponding set R_j^* to $\mathcal{R}'$ and solve ($\overline{\text{LMRP-SC}}$) again. This method is called *column generation* since it consists of generating good variables (columns) on the fly.

12.2.8 Conic Optimization

We now introduce a more general approach to solve the LMRP. The approach is based on recent developments in conic integer programming. The basic idea is to reformulate the LMRP as a *conic quadratic mixed-integer program* (CQMIP), which can then be solved directly using standard optimization software packages such as CPLEX or Mosek, without the need for specially designed algorithms, such as the column generation algorithm in Section 12.2.7 and the Lagrangian relaxation algorithm in Section 12.2.6. Moreover, it does not require us to make Assumption 12.1 about the variance-to-mean ratio at the retailers. The approach we discuss was introduced by Atamturk et al. (2012). Classical facility location problems have also been formulated and solved as conic quadratic programs (see, e.g., Kuo and Mittelmann (2004)).

Let $t_{1j}, t_{2j} \geq 0$ be auxiliary decision variables that represent the nonlinear terms in the objective function (12.8):

$$t_{1j} = \sqrt{\sum_{i \in I} \mu_i y_{ij}} \tag{12.33}$$

$$t_{2j} = \sqrt{\sum_{i \in I} \sigma_i^2 y_{ij}} \tag{12.34}$$

Then the LMRP can be reformulated as an equivalent CQMIP as follows:

(LMRP-CQMIP)

$$\text{minimize} \quad \sum_{j \in I} \left[f_j x_j + \sum_{i \in I} \mu_i (c_j + d_{ij}) y_{ij} + \sqrt{2K_j h_j} t_{1j} + h_j z_\alpha \sqrt{L_j} t_{2j} \right] \tag{12.35}$$

$$\text{subject to} \quad \sum_{i \in I} \mu_i y_{ij}^2 \leq t_{1j}^2 \qquad \forall j \in J \tag{12.36}$$

$$\sum_{i \in I} \sigma_i^2 y_{ij}^2 \leq t_{2j}^2 \qquad \forall j \in J \tag{12.37}$$

$$t_{1j}, t_{2j} \geq 0 \qquad \forall j \in J \tag{12.38}$$

$$\text{(12.9)–(12.12)}$$

The objective function is identical to (12.8) except that the nonlinear terms have been replaced by t_{1j} and t_{2j}. Constraints (12.36) and (12.37) enforce the definitions of these two variables, as given in (12.33)–(12.34). Note that, since $y_{ij} \in \{0, 1\}$, we have $y_{ij}^2 = y_{ij}$. Moreover, since t_{1j} and t_{2j} have positive coefficients in the objective function, it is sufficient to use inequalities in (12.36)– (12.37) in place of the equalities in (12.33)–(12.34).

The objective function of (LMRP-CQMIP) is linear, and the constraints are all either conic quadratic or linear, so (LMRP-CQMIP) fits the general form of a CQMIP model and can therefore be solved using a general-purpose CQMIP solver. Atamturk et al. (2012) report that the computational performance of this method is similar to or better than the Lagrangian relaxation and column generation methods discussed above.

Moreover, this approach is very flexible and can be adapted to handle other extensions of the LMRP. For example, Ozsen et al. (2008) incorporate capacity constraints into the LMRP. This is not as straightforward as it is in the capacitated fixed-charge location problem (Section 8.3.1) because the capacity applies to the maximum inventory level (the inventory level that occurs immediately after an order arrives) rather than simply to the total annual throughput, as in the CFLP. Moreover, because there is no closed-form expression for the optimal expected cost of a capacitated (r, Q) policy (as there is for an uncapacitated policy), the capacitated LMRP model requires an explicit Q_j variable to represent the order-quantity decision. The expected cycle-stock cost is expressed in the objective function as

$$K_j \frac{\sum_{i \in I} \mu_i y_{ij}}{Q_j} + h_j \frac{Q_j}{2} \tag{12.39}$$

(analogous to the EOQ cost function (3.3)), where Q_j is a decision variable. The safety-stock cost is still given by the second term in (12.6). As a result, the capacitated LMRP model, as formulated by Ozsen et al. (2008), is neither convex nor concave. Ozsen et al. (2008) propose a heuristic method based on Lagrangian relaxation to solve it. However, Atamturk et al. (2012) show that the capacitated LMRP can be reformulated as a CQMIP, using similar ideas as those given above, and solved using a general-purpose solver. They also use this approach to solve other variants of the LMRP, such as problems with correlated retailer demands, stochastic lead times, and multiple commodities.

12.3 A LOCATION–ROUTING MODEL

Location and routing decisions are closely related, since they both depend on the spatial relationships among the facilities and customers. There is often a cost savings that can be attained when the two decisions are optimized simultaneously. The model discussed in this section, which is adapted from Laporte and Nobert (1981) and Laporte et al. (1986), is only one of many location–routing models in the literature. (For other approaches, see, for example, Perl and Daskin (1985) or Berger et al. (2007).) However, because it has become a seminal model, we refer to it as “the” location–routing problem (LRP). For more thorough reviews of location–routing, see Nagy and Salhi (2007), Prodhon and Prins (2014), and Drexl and Schneider (2015).

The location–routing model we discuss combines elements of the UFLP and the vehicle routing problem (VRP). Both the UFLP and the VRP are NP-hard, which makes the integrated model even more complex. Indeed, since the VRP is a special case of the LRP (obtained by setting the fixed location costs to 0), the LRP is NP-hard as well.

The LRP aims to make three sets of related decisions: which depots to open, which customers to assign to each depot, and how to route the vehicles from each depot to its customers. We assume that the number of vehicles is finite and that each vehicle has a (possibly finite) capacity.

We use the following notation, which borrows from the notation of both the UFLP (Section 8.2.2) and the VRP (Section 11.1.3):

Sets

I = set of customers
J = set of potential depot locations
N = set of all nodes: $N = I \cup J$
E = set of undirected edges between nodes: $E = \{(i,j)|i,j \in N\}$

Parameters

Demand and Capacity

d_i = demand of customer $i \in I$
D = capacity of each vehicle

Costs

f_j = fixed cost to open a depot at node $j \in J$
g_j = fixed cost per vehicle used at depot $j \in J$
c_{ij} = cost to transport one unit of demand along edge $(i,j) \in E$; defined only for $i < j$

Other

M = an arbitrarily large number
p = pre-specified upper bound on the total number of open depots
$\overline{m}_j$ = pre-specified upper bound on the number of vehicles assigned to depot $j \in J$

Decision Variables

y_j = 1 if a depot is opened at node $j \in J$, 0 otherwise
x_{ij} = the number of times edge $(i, j) \in E$ is used; x_{ij} is not defined if $i \geq j$, if i and j are both in J, or if $d_i + d_j > D$
m_j = number of vehicles assigned to depot $j \in J$

The objective of the LRP is to minimize the sum of three terms: the fixed cost of opening depots, the fixed cost of using vehicles, and the transportation cost from the vehicle routes. A feasible solution must satisfy the following constraints: each vehicle route begins and ends at the same depot; each vehicle performs at most one trip; each customer is served by a single vehicle; and the total demand of customers visited by each vehicle does not exceed its capacity.

$$\text{(LRP)} \quad \text{minimize} \quad \sum_{i,j \in I} c_{ij} x_{ij} + \sum_{j \in J} (f_j y_j + g_j m_j) \tag{12.40}$$

$$\text{subject to} \quad \sum_{i \in N} x_{ik} + \sum_{j \in N} x_{kj} = 2 \qquad \forall k \in I \tag{12.41}$$

$$\sum_{i \in I} x_{ik} + \sum_{j \in I} x_{kj} = 2m_k \qquad \forall k \in J \tag{12.42}$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - \left\lceil \frac{\sum_{k \in S} d_k}{D} \right\rceil \qquad \forall S \subseteq I : |S| \geq 3 \tag{12.43}$$

$$x_{i_1 i_2} + 3x_{i_2 i_3} + x_{i_3 i_4} \leq 4 \qquad \forall i_1, i_4 \in J, \forall i_2, i_3 \in I \tag{12.44}$$

$$x_{i_1 i_2} + x_{i_{h-1} i_h} + 2 \sum_{i,j \in \{i_2, \ldots, i_{h-1}\}} x_{ij} \leq 2h - 5 \qquad \forall h \geq 5, \forall i_1, i_h \in J, \forall i_2, \ldots, i_{h-1} \in I \tag{12.45}$$

$$y_j \leq m_j \leq M y_j \qquad \forall j \in J \tag{12.46}$$

$$0 \leq m_j \leq \overline{m}_j \qquad \forall j \in J \tag{12.47}$$

$$1 \leq \sum_{j \in J} y_j \leq p \tag{12.48}$$

$$y_j \in \{0, 1\} \qquad \forall j \in J \tag{12.49}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i, j \in I \tag{12.50}$$

$$x_{ij} \in \{0, 1, 2\} \qquad \forall i, j \in N : i \text{ or } j \in J \tag{12.51}$$

Here, $\lceil t \rceil$ is a function that equals the smallest positive integer that is greater than or equal to t. As in the formulation of the TSP (Section 10.2.2), we have not bothered to add the requirement that $i < j$ every time x_{ij} appears, but this should be understood—for example, in the first summation in (12.40), in the summations in (12.41) and (12.42), in the "for all" part of (12.44), and so on.

The objective function (12.40) calculates the total transportation cost plus the total fixed costs of opening depots and using vehicles. Constraints (12.41) are degree constraints, specifying that the total number of edges connected to a customer node $k \in I$ is 2.

Constraints (12.42) are also degree constraints, this time for the depot nodes: The number of edges connected to a depot must equal twice the number of vehicles used at that depot. Note that the sums are over the set I rather than N since the depot cannot be connected by edges to other depots. Constraints (12.43) are capacity-cut constraints, analogous to (11.4) and (11.9) for the VRP, eliminating subtours and ensuring capacity feasibility of the routes. Constraints (12.44) and (12.45) are called *chain-barring constraints*, which ensure that each route starts and ends at the same depot. We refer to Laporte et al. (1986) for details of the development of these constraints. Constraints (12.46) are linking constraints, enforcing the relationship between m_j and y_j so that no vehicles can be based at a node that is not selected as a depot. Constraints (12.47) and (12.48) impose bounds on the number of vehicles based at node j and the total number of nodes used as depots. Constraints (12.49)–(12.51) are integrality constraints; note that $x_{ij} = 2$ is allowed if i or j is a depot, corresponding to a round trip directly between i and j.

This formulation of the LRP contains exponentially many constraints in (12.43)–(12.45), and hence, it is not easy to solve this problem by directly tackling the explicit version as an integer program. Laporte et al. (1986) propose an exact algorithm using constraint generation, in which the capacity-cut and chain-barring constraints are first removed from the model, and then violated constraints are added back, all within a branch-and-bound framework. Berger et al. (2007) consider a variant of (12.43)–(12.45), which they reformulate as a set partitioning problem and solve using column generation.

12.4 AN INVENTORY–ROUTING MODEL

Inventory–routing problems combine the VRP with inventory management problems. Their development was sparked, in part, by the rise of *vendor-managed inventory* (VMI) arrangements, in which suppliers (vendors) take responsibility for replenishing the inventory of their products at their customers (that is, at retailers). VMI can be effective in reducing logistics costs, inventory levels, and the bullwhip effect (see Section 13.3). This brings about a new challenge for the vendor, which must now route vehicles to deliver products while also keeping an eye on customers' inventory levels.

Like location–inventory and location–routing, inventory–routing is really a class of problems, rather than a single problem. In this section, we discuss a seminal version, which we will refer to as "the" inventory–routing problem (IRP). We refer interested readers to Coelho et al. (2013) for a detailed review. Inventory–routing problems are especially common in maritime logistics, with applications in the chemical, oil, and gas industries, as well as a wide range of consumer and business goods in both maritime and non-maritime distribution systems.

The IRP considers three sets of decisions: when to deliver to each customer, how much to deliver, and how to route vehicles to the customers. The objective is to minimize the total inventory holding cost and distribution cost, subject to constraints on the inventory levels at customers and the feasibility of the vehicle routes. Stockouts are not allowed. The model is dynamic in the sense that it considers the movements of vehicles and inventory over time, rather than the static approach taken by the LMRP and the LRP. We assume that the time required for the vehicles to complete their routes is small compared to the length of one time period. Since the VRP is a special case of the IRP, the IRP is NP-hard.

Our formulation for the IRP will include an index k on the decision variables to indicate the vehicle. In this sense, it is similar to the three-index formulation of the VRP (see Section 11.1.3), whereas the formulation of the LRP in Section 12.3 was similar to the two-index VRP formulation. Note, however, that the IRP variables will also have an extra index for the time period. We will use the following notation:

Sets

I	= set of customers
N	= set of all nodes: 0 is the depot and $N = I \cup \{0\}$
E	= set of undirected edges between nodes: $E = \{(i,j) \mid i,j \in N\}$
T	= set of time periods; $T = \{1, \ldots, T\}$
K	= set of vehicles

Parameters

Supply, Demand, and Capacity

r_t	= new supply available at the depot in period $t \in T$
d_{it}	= demand of customer $i \in I$ in period $t \in T$
C_i	= inventory holding capacity of customer $i \in I$
D_k	= capacity of vehicle $k \in K$

Costs

h_i	= holding cost per unit of inventory at node $i \in N$ per period
c_{ij}	= cost to transport one unit of demand along $(i,j) \in E$; defined only for $i < j$

Other

I_{i0}	= inventory level at node $i \in N$ at the beginning of the planning horizon

Decision Variables

x_{ijkt}	= the number of times edge $(i,j) \in E$ is used by vehicle $k \in K$ in period $t \in T$; defined only for $i,j \in N$ such that $i < j$ and $d_i + d_j \le D_k$
y_{ikt}	= 1 if vehicle $k \in K$ visits node $i \in N$ in period $t \in T$, 0 otherwise
q_{ikt}	= number of units delivered to customer $i \in I$ by vehicle $k \in K$ in period $t \in T$
I_{it}	= inventory level at node $i \in N$ at the end of period $t \in T$

The IRP can now be formulated as a mixed-integer programming problem as follows:

(IRP)

$$\text{minimize} \quad \sum_{i \in N} \sum_{t \in T} h_i I_{it} + \sum_{(i,j) \in E} \sum_{k \in K} \sum_{t \in T} c_{ij} x_{ijkt} \tag{12.52}$$

$$\text{subject to} \quad I_{0t} = I_{0,t-1} + r_t - \sum_{k \in K} \sum_{i \in I} q_{ikt} \qquad \forall t \in T \tag{12.53}$$

$$I_{0t} \ge 0 \qquad \forall t \in T \tag{12.54}$$

$$I_{it} = I_{i,t-1} + \sum_{k \in K} q_{ikt} - d_{it} \qquad \forall i \in I, \forall t \in T \tag{12.55}$$

$$0 \le I_{it} \le C_i \qquad \forall i \in I, \forall t \in T \tag{12.56}$$

$$\sum_{k \in K} q_{ikt} \le C_i - I_{i,t-1} \qquad \forall i \in I, \forall t \in T \tag{12.57}$$

$$0 \le q_{ikt} \le C_i y_{ikt} \qquad \forall i \in I, \forall k \in K, \forall t \in T \tag{12.58}$$

$$\sum_{i \in I} q_{ikt} \le D_k y_{0kt} \qquad \forall k \in K, t \in T \tag{12.59}$$

$$\sum_{j \in N} x_{ijkt} + \sum_{j \in N} x_{jikt} = 2y_{ikt} \qquad \forall i \in N, \forall k \in K, \forall t \in T \tag{12.60}$$

$$\sum_{i,j \in S} x_{ijkt} \le \sum_{i \in S} y_{ikt} - y_{mkt} \qquad \forall S \subseteq I, \forall k \in K, \forall t \in T, \forall m \in S \tag{12.61}$$

$$x_{i0kt} \in \{0, 1, 2\} \qquad \forall i \in I, \forall k \in K, \forall t \in T \tag{12.62}$$

$$x_{ijkt} \in \{0, 1\} \qquad \forall i, j \in I, \forall k \in K, \forall t \in T \tag{12.63}$$

$$y_{ikt} \in \{0, 1\} \qquad \forall i \in N, \forall k \in K, \forall t \in T \tag{12.64}$$

Note that, as in the LRP, we are omitting the condition $i < j$ from the formulation, but it should be understood whenever x_{ijkt} or c_{ij} is present. The objective function (12.52) calculates the total inventory-holding and transportation cost for every node, every vehicle, and every time period. Constraints (12.53) define the dynamics of the inventory level of the depot: The inventory level at the end of period t equals the inventory level at the end of period $t-1$, plus the new supply available, minus the quantity shipped to customers. Constraints (12.54) ensure that the depot never stocks out. Constraints (12.55) similarly define the inventory dynamics at customers: The inventory at the end of period t equals the inventory at the end of period $t-1$, plus the number of delivered items, minus the demand. Constraints (12.56) ensure that there are no stockouts at the customers and that the inventory does not exceed the storage capacity. Note that these constraints assume that the *ending* inventory does not exceed the capacity, meaning that the capacity might be temporarily violated before all of the demand has occurred in each time period; this assumption is typical in IRP models. Constraints (12.57) prevent the total amount delivered to customer i by all vehicles in period t from exceeding the available storage capacity. Constraints (12.58) enforce the relationship between the quantities delivered, q_{ikt}, and the routing variables y_{ikt}: The quantity delivered to customer i by vehicle k in period t must be 0 if $y_{ikt} = 0$; if $y_{ikt} = 1$, then the quantity delivered may not exceed the capacity. Constraints (12.59) ensure that vehicle capacities are not exceeded. Constraints (12.60) and (12.61) are degree constraints and subtour-elimination constraints, respectively (similar to constraints (11.13) and (11.15) for the VRP). Constraints (12.62)–(12.64) are integrality constraints.

This formulation contains exponentially many subtour-elimination constraints (12.61). In practice, it is usually solved by removing these constraints and adding violated constraints back to the branch-and-bound tree in a constraint generation scheme, similar to the LRP; see Archetti et al. (2007), Coelho and Laporte (2013), and Adulyasak et al. (2014) for details.

As we mentioned above, the IRP formulation (12.52)–(12.64) is only one of many inventory–routing problems that have been proposed. Other such problems differ in terms of various criteria such as time horizon (single or multiple period), inventory policy (maximum-level or order-up-to), handling of stockouts (backorders or lost sales), and fleet composition (homogeneous or heterogeneous vehicles). Moreover, this basic form of the IRP has been extended to include variants such as the production–routing problem (in which we make production decisions at the depot), the IRP with multiple products, the IRP with direct deliveries and transshipments, the multi-item IRP, the IRP with several suppliers and customers, and the IRP with heterogeneous fleets. If the customer demands are stochastic, then we have the stochastic IRP (SIRP). In the SIRP, inventory shortages may occur and a penalty cost is imposed whenever a customer experiences a stockout. The objective is to minimize the expected (or sometimes worst-case) total inventory and transportation cost. In some versions of the problem, the customer demand is gradually revealed over time, in which case we must solve the problem repeatedly with updated demand information; this is called the dynamic and stochastic IRP (DSIRP). See Coelho et al. (2013) for further discussion of these variants.

CASE STUDY 12.1 Inventory–Routing at Frito-Lay

Frito-Lay is a very large manufacturer and distributor of snack foods. Its North American operations include over 30 factories, hundreds of DCs, and more than 20,000 trucks. The company participates in vendor-managed inventory (VMI) programs in which it is responsible for managing the inventory at retailers and other downstream locations, as well as for delivering to those locations. As we noted in Section 12.4, the inventory–routing problem (IRP) is commonly used in VMI settings. Frito-Lay's supply chain, however, is more complex than the system modeled by the IRP in Section 12.4, so the company worked with researchers at Texas A&M University to build an appropriate model and algorithm. This research is described by Çetinkaya et al. (2009).

Their model considers a single factory, from which products are shipped to DCs, distribution points called "bins," and other factories. Most customers are served by the DCs and bins, but some larger customers (called "direct-delivery" (DD) customers) are served directly from the factory. Shipments from the factory are made using large trucks that make multiple stops, and therefore their routes must be optimized. On the other hand, shipments from the DCs to non-DD customers are made on smaller trucks that travel directly to the customer and back; no routing optimization is performed by the model.

The Frito-Lay model presented by Çetinkaya et al. (2009) differs from the IRP in Section 12.4 in several ways. The most notable difference is the DC echelon, which does not exist in the classical IRP. The Frito-Lay model allows inventory to be stored at the factory and at the DCs and bins, but not at the customers. (In contrast, the IRP allows inventory at the customers.) It makes production decisions at the factory, an aspect that is considered an input in the IRP. It also considers multiple products, which have separate demands but which share production, storage, and shipment capacities. It considers fixed costs for each shipment, and transportation costs are assessed per mile, regardless of the load of the vehicle on those miles.

The resulting MIP model cannot be solved exactly using off-the-shelf solvers. Instead, Çetinkaya et al. (2009) propose a heuristic that iterates between two subproblems, which solve the inventory and routing aspects of the problem. The routing problem optimizes truck routes using the Clarke–Wright savings heuristic (see Section 11.3.1) and an improvement heuristic based on cheapest insertion (Section 10.4.2). The inventory subproblem determines the production quantities at the factory and the weekly replenishment and shipment quantities at the DCs, bins, and DD customers, accounting for the routing costs that are an output of the routing subproblem. The inventory subproblem is solved as a MIP using CPLEX.

Çetinkaya et al. (2009) report that their heuristic executes in about 10 minutes. They compare the results of their heuristic with two benchmark policies that are meant to mimic Frito-Lay's typical decision-making process and find that their solutions are better by 6%–11%. Compared to the benchmark policies, the solutions from the heuristic have lower inventory levels, lower handling and transportation costs, and comparable vehicle utilizations.

PROBLEMS

12.1 (LR Iteration for LMRP) The file `LR-LMRP.xlsx` contains a_i and b_i values for a 50-node instance of problem (P'_j) ((12.19)–(12.20)) for a single iteration of the Lagrangian relaxation algorithm described in Section 12.2.6 and for a single value of j. Using the algorithm described in Section 12.2.6.1, solve this instance of problem (P'_j). List the optimal values of y^* in column `D` and the optimal objective value (β_j) in cell `H2`.

12.2 (Nonconvexity of S_r in LMRP Algorithm) Theorem 12.1 allows us to solve problem (P'_j) by first sorting a subset of the i's and then computing the partial sums given by (12.23), choosing the r that minimizes S_r and setting $y_i = 1$ for $i = 1, \ldots, r$. It is tempting to think that S_r is convex with respect to r, since then we could consider each r in turn as long as S_r is decreasing, and then stop as soon as S_r increases (or use an even more efficient method like binary search). Unfortunately, this claim is *not* true. Provide a counterexample with four variables such that $a_i < 0$ and $b_i > 0$ for all i, and such that

$$S_1 > S_2 < S_3 > S_4 \qquad \text{and} \qquad S_2 > S_4.$$

12.3 (Retailer Assignment is NP-Hard) Suppose the facility locations in the LMRP are already fixed. Prove that the problem of optimally assigning retailers to facilities is NP-hard.

12.4 (Alternate Proof of Theorem 12.1?) A student once proposed the following approach to prove Theorem 12.1, part 3: Instead of creating two new solutions y' and y'' as in the proof, just create one new solution, defined as follows:

$$y'_i = \begin{cases} 1, & \text{if } i = \ell \\ 0, & \text{if } i = k \\ y^*_i, & \text{otherwise} \end{cases}$$

Then, prove that $z' < z^*$, where z' and z^* are the objective function values of the solutions y' and y^*, respectively. This would contradict the assumption that y^* is optimal and thus complete the proof.

Show that this approach *does not work* by creating a set I_1^- (with $a_i < 0$, $b_i > 0$, and the elements sorted in increasing order of a_i/b_i) and a solution y^* such that:

- $|I_1^-| \leq 5$
- $y^*_k = 1$, $y^*_\ell = 0$ for some $\ell < k$
- If we define y' as above, then $z' > z^*$.

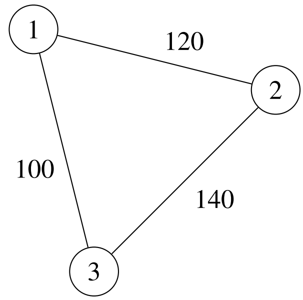


Figure 12.5 3-Node LMRP instance for Problem 12.5.

12.5 (Nonclosest Assignments in LMRP) Consider the 3-node instance of the LMRP pictured in Figure 12.5. Each circle represents a retailer, and each retailer is eligible to be converted to a DC. The numbers on the links indicate the transportation cost (d_{ij}) between retailers; the transportation cost between a retailer and itself is 0. Construct an example, using this instance, for which there is a retailer in the optimal solution that is assigned to a DC that is not its closest open DC. That is, this problem is asking you to choose values for the parameters (f_j, μ_i, σ_i, K_j, etc., but not including d_{ij} since those are given in the diagram) and demonstrate that in the optimal solution, there is a retailer assigned to a given DC, even though there is another open DC that is closer. You do not need to enumerate the objective value of every possible solution, but you should argue rigorously that your candidate solution is optimal.

12.6 (LRP with Distance Constraints) In this problem, you will extend the location–routing problem (LRP) to consider distance constraints on the length of the vehicle routes, and you will reformulate the model as a set partitioning problem. In addition to the notation in Section 12.3, let ℓ_{ij} be the distance between nodes i and j, and let $\bar{\ell}$ be a prespecified upper bound on the total distance of each route. Let $\mathcal{R}_j$ be the set of all feasible routes from depot $j \in J$. As in Section 12.3, a route is only feasible if it begins and ends at the same depot and does not violate its capacity, and now we also require it to satisfy the distance constraint. Let c_{jR} be the cost of route $R \in \mathcal{R}_j$, which equals the sum of the costs c_{ij} of the edges on the route. Let a_{ijR} be a parameter that equals 1 if route $R \in \mathcal{R}_j$ visits customer i, and 0 otherwise.

The model aims to choose one route $R \in \mathcal{R}_j$ for each $j \in J$ to minimize the total cost of the routes. Every customer must be included in exactly one route. The objective function consists of the total fixed cost of opening depots plus the total routing cost. As before, y_j is a decision variable that equals 1 if depot $j \in J$ is opened, and now let z_{jR} be a new decision variable that equals 1 if route $R \in \mathcal{R}_j$ is selected and 0 otherwise. You may assume there is no upper bound on the number of depots open or the number of vehicles assigned to each depot.

a) Formulate the LRP with distance constraints as a linear integer programming problem. If you introduce any new notation, define it clearly. Explain the objective function and constraints in words.

b) Design a heuristic for the LRP with distance constraints that is based on the greedy-add heuristic we discussed for the UFLP in Section 8.2.5. Write pseudocode for your heuristic. You may assume that your heuristic has access to the following two black-box functions and may use them as many times as you wish. ("Black-box" means you don't know how the functions work, but you may call them and use their results.)

- `Route_Enumerate(j,I)` takes as input a candidate depot location j and a customer set I and returns as output the set $\mathcal{R}_j$ of all feasible routes for depot j. The function also returns, for each route $R \in \mathcal{R}_j$, the values of the parameters c_{jR} and a_{ijR}.

- `Route_Select(j,I)` takes as input a candidate depot location j and a customer set I and returns as output a set $\hat{\mathcal{R}}_j \subseteq \mathcal{R}_j$, which contains the lowest-cost subset of all feasible routes for depot j such that every customer in I is visited by exactly one route. The function also returns, for each route $R \in \hat{\mathcal{R}}_j$, the values of the parameters c_{jR} and a_{ijR}.

Note that in both functions, the input parameter I may equal the entire customer set or only a subset of it.

CHAPTER 13

THE BULLWHIP EFFECT

13.1 INTRODUCTION

In the early 1990s, executives at Procter & Gamble (P&G) noticed a peculiar trend in the orders for Pampers, a brand of baby diapers. As you might expect, demand for diapers at the consumer level is pretty steady since babies use them at a fairly constant rate. But P&G noticed that the orders placed by retailers (e.g., CVS, Target) to distributors were quite variable over time—high one week, low the next. The distributors' orders to P&G were even more variable, and P&G's orders to its own suppliers (e.g., 3M) were still more variable. (See Figure 13.1.)

This phenomenon is known as the *bullwhip effect* (BWE), a phrase coined by P&G executives that refers to the way a wave's amplitude increases as it travels the length of a whip. Sometimes it's also known as the "whiplash" or "whipsaw" effect. The BWE has been observed in many industries other than diapers. For example, Hewlett-Packard (HP) noticed large variability in the orders retailers placed to HP for printers, even though demand for printers is fairly steady. Similarly, the demand for DRAM (a component of computers) is more volatile than the demand for computers themselves. Wide swings in order sizes can cause big increases in inventory costs (for both raw materials and finished goods), overtime and idling expenses, and emergency shipment costs. These factors are estimated to increase costs by as much as 12.5–25% (Lee et al. 1997a).

The BWE was described in the literature as early as the 1950s (Forrester 1958). Sterman (1989) described how the BWE could be caused by *irrational* behavior by supply chain

Fundamentals of Supply Chain Theory, Second Edition. Lawrence V. Snyder and Zuo-Jun Max Shen.

Companion website: www.wiley.com/go/Snyder/SupplyChainTheory

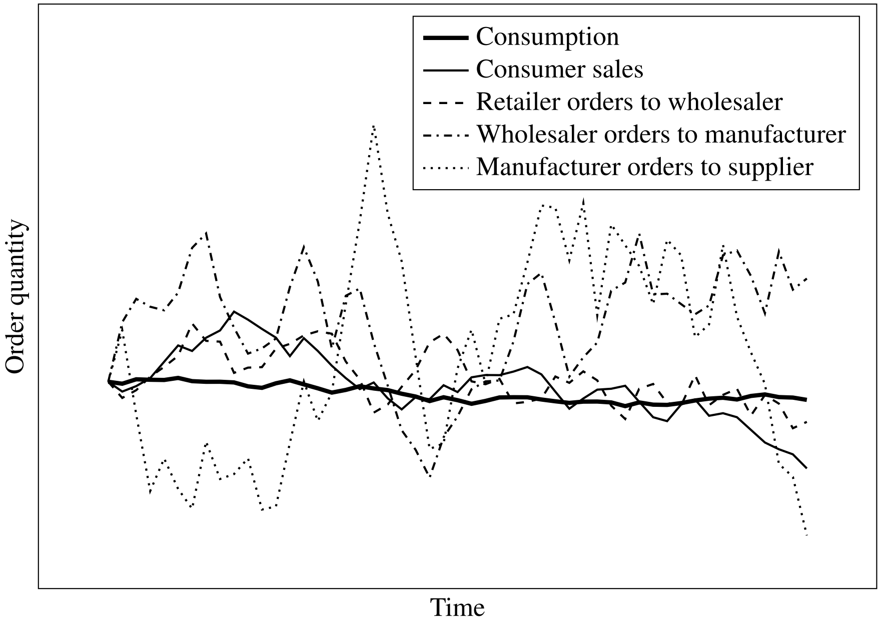


Figure 13.1 Increase in order variability in upstream supply chain stages.

managers: for example, overreacting to a small shortage one week by ordering far too much the next week. His paper uses the now-famous "beer game" to demonstrate this relationship empirically. Then, two papers by Lee et al. (1997a,b) demonstrated that the BWE can occur even if all players act *rationally*—following the logical, optimized policies of the type we discuss in this book. They identified four primary causes for the BWE:

1. **Demand signal processing.** Many firms use forecasting techniques to estimate the mean and standard deviation of current or future demands. Each time a new demand is observed, the estimates are updated. If the previous period's demand was high, the new estimate will be higher than the previous one, thus raising the target inventory level. The orders will be more exaggerated than the demands. This phenomenon is amplified by the lead time if the base-stock level is set using (4.46), that is,

$$\mu L + z_\alpha \sigma \sqrt{L},$$

   where μ and σ are the mean and standard deviation of the demand per period and L is the lead time. In practice, the firm doesn't know μ and σ, so it estimates them based on historical data. These estimates change periodically, and any change in the estimates are magnified by the lead time when setting base-stock levels, so long lead times produce large shifts in order sizes.

2. **Rationing game.** When distributors don't have enough inventory to meet retailers' orders, they often allocate product according to order size: If retailer 1 orders 100 units and retailer 2 orders 150 units, but there are only 200 units available, then the distributor will give $200 \times (100/250) = 80$ units to retailer 1 and $200 \times (150/250) = 120$ units to retailer 2. If the retailers anticipate the shortage, they may artificially

inflate their orders to try to get a larger allocation of the inventory. Once the shortage is over, the retailers' orders will return to their normal levels, or even lower. Thus, the variance in retailer orders is larger than the variance in actual demands.

3. **Order batching.** It is common for all players in a supply chain to place orders in bulk: Parents buy diapers in packages of 50, retailers buy them by the case, distributors buy them by the truckload. A theoretical explanation for the optimality of bulk ordering is that there is a fixed cost to place each order, so it's better to place fewer orders if possible. (For parents, the fixed cost is in the form of inconvenience and time: They don't want to go to the drug store every time their baby uses another diaper.) Moreover, bulk buying is encouraged by sellers by offering quantity discounts, another common practice. But order batching means that orders may be high one week, then low for the next few weeks as retailers use up the stock they've accumulated. Another reason for order batching is that many firms use material requirements planning (MRP) software that evaluates the firm's requirements for every part it uses and automatically places orders with suppliers once per month. That means the supplier sees large demand during a few days of the month as its customers' MRP systems place orders and small demand for the rest of the month; this is sometimes known as the "hockey stick" phenomenon.

4. **Price speculation.** Prices change all the time, and firms tend to stock up while prices are low and order less when prices are high. This is most pronounced at upstream stages in the supply chain, whose raw materials are commodities such as plastic, steel, fuel, and so on—prices for these commodities change constantly, and speculation is common among buyers. It's also obvious at the other end of the supply chain, as customers buy more when retailers offer sales and promotions. In the middle of the supply chain, sales and promotions are common, too, causing retailers and other players to stock up when prices drop. All of this leads to large variability in buying patterns.

In Section 13.2, we'll discuss mathematical models explaining these causes and demonstrating that they occur even when each player in the supply chain is a rational "optimizer." Then, in Section 13.3, we'll discuss strategies for reducing the BWE. Finally, in Section 13.4, we'll examine the extent to which sharing demand information with upstream supply chain members can reduce or eliminate the BWE.

Most of the analysis in this section is adapted from Lee et al. (1997a) and Chen et al. (2000). For reviews of the literature on the BWE, see McCullen and Towill (2002), Lee et al. (2004), or Geary et al. (2006).

13.2 PROVING THE EXISTENCE OF THE BULLWHIP EFFECT

13.2.1 Introduction

Consider a serial supply chain like the one pictured in Figure 13.2. We will examine this system in the context of an infinite horizon under periodic review. Each stage places orders from its upstream stage and supplies product to its downstream stage. Stage N serves the end customer.

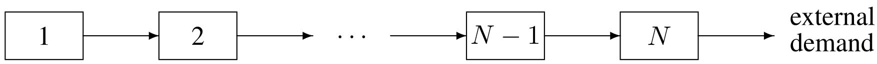


Figure 13.2 Serial supply chain network.

Our strategy will be to focus on one stage and to show that the variance of orders it places to its supplier is larger than the variance of orders it receives from its customer. That, in turn, implies the BWE as a whole: Stage N's orders are more variable than its demands, so stage $N-1$'s orders are even more variable, so stage $N-2$'s orders are even more variable, and so on.

Suppose the following conditions hold at each stage:

1. Demands are independent over time, and the parameters of the demand distribution are known.

2. The stage's supplier (i.e., its upstream stage) always has sufficient inventory and satisfies orders with a fixed lead time that is independent of the order size.

3. There is no fixed ordering cost.

4. The purchase cost is constant over time.

If all four of these conditions hold, it is optimal for the stage to follow a stationary base-stock policy. As we know from Section 4.3, that means that in each period, the order placed by the stage is exactly equal to the demand seen by the stage in the previous time period, so the orders placed by the stage and the demand seen by it have the same variance—*the bullwhip effect does not occur*.

However, relaxing each of the conditions given above (one at a time) gives us the four causes of the BWE: demand signal processing (when the demand parameters are unknown and hence forecasting techniques must be used to estimate them), rationing game (when supply is limited), order batching (when there is a fixed cost for ordering), and price speculation (when the purchase price can change over time).

We discuss models for each of these causes next. In each of the four sections that follows, we will consider only a single stage in the supply chain and show that the orders placed by the stage to its supplier have larger variance than the demands received by the stage. Without loss of generality we will refer to this stage as the "retailer."

13.2.2 Demand Signal Processing

In this section, we relax both parts of assumption #1 in Section 13.2.1: We assume that the demands are *serially correlated*—that is, demands in one time period are statistically dependent on demands in the previous time period—and that the parameters of the demand process are unknown and must be estimated. Each stage in the supply chain makes its own estimate of the demand parameters based on the orders it receives. We will show that this processing of the demand signal can lead to the BWE.

We assume that the demands seen by the retailer follow a first-order autoregressive $AR(1)$ process; that is, the demand follows a model of the form

$$D_t = d + \rho D_{t-1} + \epsilon_t, \qquad (13.1)$$

where D_t is the demand in period t (a random variable), $d \geq 0$ is a constant, ρ is a correlation constant with $-1 < \rho < 1$, and ϵ_t is an error term that is distributed $N(0, \sigma^2)$. If ρ is close to 1, then a large demand tends to be followed by another large demand, while if ρ is close to -1, then a large demand tends to be followed by a small one.

It's tempting to think of d as the mean of this process, but it is not, unless $\rho = 0$. In fact, it can be shown that

$$\mathbb{E}[D_t] = \frac{d}{1-\rho} \qquad (13.2)$$

$$\mathrm{Var}[D_t] = \frac{\sigma^2}{1-\rho^2} \qquad (13.3)$$

$$\mathrm{Cov}[D_t, D_{t-k}] = \frac{\rho^k \sigma^2}{1-\rho^2} = \rho^k \mathrm{Var}[D_t]. \qquad (13.4)$$

Note that the mean, variance, and covariance are the same in every period. If $\rho = 0$, the demands are iid with mean d and variance σ^2. These are steady-state values; if we know D_{t-1}, then these formulas do not apply. (See, for example, Problem 13.3.)

The retailer follows a base-stock policy. Let D_t^L be the lead-time demand for an order placed in period t; that is,

$$D_t^L = \sum_{k=0}^{L-1} D_{t+k}. \qquad (13.5)$$

If the retailer knew the mean $\mu_t^L = \mathbb{E}[D_t^L]$ and standard deviation $\sigma_t^L = \sqrt{\mathrm{Var}[D_t^L]}$ of the lead-time demand (which it could calculate if it knew d, σ, and ρ—see Problem 13.3), then, analogous to (4.46), the optimal base-stock level would be given by

$$S_t = \mu_t^L + z_\alpha \sigma_t^L. \qquad (13.6)$$

However, the retailer does not know μ_t^L and σ_t^L but instead must forecast them based on observed demands using, for example, one of the methods in Chapter 2. One of the most common forecasting techniques, and the one we'll use here, is a *moving average* (Section 2.2.1), which simply consists of the average of the demands from the previous m time periods. The estimate for μ_t^L, computed at time t and denoted $\hat{\mu}_t^L$, is

$$\hat{\mu}_t^L = L\left(\frac{\sum_{i=1}^m D_{t-i}}{m}\right). \qquad (13.7)$$

As for the standard deviation, it turns out that instead of estimating the standard deviation of lead-time demand (σ_t^L), we want to estimate the standard deviation of the *forecast error* of the lead-time demand, σ_{et}^L. (See Section 4.3.2.7.) The estimate of σ_{et}^L at time t is given by

$$\hat{\sigma}_{et}^L = C_{L\rho}\sqrt{\frac{\sum_{i=1}^m (e_{t-i})^2}{m}} \qquad (13.8)$$

where

$$e_t = D_t - \hat{\mu}_t^1$$

is the one-period forecast error and $C_{L\rho}$ is a constant depending on L, ρ, and m; we omit the derivation of this equation and the exact form of $C_{L\rho}$. The base-stock level is then set using

$$S_t = \hat{\mu}_t^L + z_\alpha \hat{\sigma}_{et}^L. \tag{13.9}$$

This policy is optimal for iid normal demands (i.e., if $\rho = 0$) and is approximately optimal otherwise. (It is only approximately optimal because these estimates of μ_t^L and σ_{et}^L do not take into account the autocorrelation of the demand; that is, they assume that the demand will have the same distribution in each period of the lead time. It would be more accurate to account for the correlation, i.e., using (13.1), when estimating the lead-time demand parameters. This is relatively straightforward to do if d, ρ, and σ are known—see Problem 13.3—but is quite a bit harder when the parameters are unknown and are estimated as described above.)

In period t, the retailer computes $\hat{\mu}_t^L$ and $\hat{\sigma}_{et}^L$ using the previous m periods' demands, then sets the base-stock level S_t using (13.9) and places an order of size $Q_t = S_t - S_{t-1} + D_{t-1}$ (why?). (It is possible that $Q_t < 0$. In this case, we assume that the firm *returns* $-Q_t$ units to the supplier and receives a full refund for the returned units.) We can write Q_t as

$$\begin{aligned}
Q_t &= S_t - S_{t-1} + D_{t-1} \\
&= \hat{\mu}_t^L + z_\alpha \hat{\sigma}_{et}^L - (\hat{\mu}_{t-1}^L + z_\alpha \hat{\sigma}_{e,t-1}^L) + D_{t-1} \\
&= \hat{\mu}_t^L - \hat{\mu}_{t-1}^L + z_\alpha (\hat{\sigma}_{et}^L - \hat{\sigma}_{e,t-1}^L) + D_{t-1} \\
&= L\left(\frac{\sum_{i=1}^m D_{t-i} - \sum_{i=1}^m D_{t-1-i}}{m}\right) + D_{t-1} + z_\alpha(\hat{\sigma}_{et}^L - \hat{\sigma}_{e,t-1}^L) \\
&= L\left(\frac{D_{t-1} - D_{t-m-1}}{m}\right) + D_{t-1} + z_\alpha(\hat{\sigma}_{et}^L - \hat{\sigma}_{e,t-1}^L) \\
&= \left(1 + \frac{L}{m}\right) D_{t-1} - \frac{L}{m} D_{t-m-1} + z_\alpha(\hat{\sigma}_{et}^L - \hat{\sigma}_{e,t-1}^L).
\end{aligned}$$

We want to compute $\mathrm{Var}[Q_t]$ so that we can compare it to $\mathrm{Var}[D_t]$ to demonstrate the BWE. Using the fact that

$$\mathrm{Var}[aX + bY] = a^2\mathrm{Var}[X] + b^2\mathrm{Var}[Y] + 2ab\mathrm{Cov}[X, Y], \tag{13.10}$$

we have

$$\begin{aligned}
\mathrm{Var}[Q_t] =& \mathrm{Var}\left[\left(1 + \frac{L}{m}\right) D_{t-1} - \frac{L}{m} D_{t-m-1}\right] + \mathrm{Var}\left[z_\alpha(\hat{\sigma}_{et}^L - \hat{\sigma}_{e,t-1}^L)\right] \\
&+ 2\mathrm{Cov}\left[\left(1 + \frac{L}{m}\right) D_{t-1} - \frac{L}{m} D_{t-m-1}, z_\alpha(\hat{\sigma}_{et}^L - \hat{\sigma}_{e,t-1}^L)\right].
\end{aligned} \tag{13.11}$$

Let's examine the $\mathrm{Cov}[\cdot]$ term. Recall that

$$\mathrm{Cov}\left[\sum_{i=1}^m a_i X_i, \sum_{j=1}^n b_j Y_j\right] = \sum_{i=1}^m \sum_{j=1}^n a_i b_j \mathrm{Cov}[X_i, Y_j].$$

Then

$$\mathrm{Cov}[\cdot] = \left(1 + \frac{L}{m}\right) z_\alpha \mathrm{Cov}[D_{t-1}, \hat{\sigma}_{et}^L] - \left(1 + \frac{L}{m}\right) z_\alpha \mathrm{Cov}[D_{t-1}, \hat{\sigma}_{e,t-1}^L]$$

$$-\frac{L}{m}z_\alpha \mathrm{Cov}[D_{t-m-1},\hat{\sigma}^L_{et}] + \frac{L}{m}z_\alpha \mathrm{Cov}[D_{t-m-1},\hat{\sigma}^L_{e,t-1}]. \tag{13.12}$$

To evaluate this further, we'll need the following lemma:

Lemma 13.1 $\mathrm{Cov}[D_{t-i},\hat{\sigma}^L_{et}] = 0$ *for all* $i = 1,\ldots,m$.

Proof. Omitted; see Ryan (1997). ■

Therefore, the first and last terms of (13.12) are equal to 0. As for the middle terms,

$$\begin{aligned}
&-\left(1+\frac{L}{m}\right) z_\alpha \mathrm{Cov}[D_{t-1},\hat{\sigma}^L_{e,t-1}] \\
=&-\left(1+\frac{L}{m}\right) z_\alpha \mathrm{Cov}[d+\rho D_{t-2}+\epsilon_{t-1},\hat{\sigma}^L_{e,t-1}] && \text{(by (13.1))}\\
=&-\left(1+\frac{L}{m}\right) z_\alpha \rho \mathrm{Cov}[D_{t-2},\hat{\sigma}^L_{e,t-1}] && \text{(since } D, \epsilon \text{ are independent)}\\
=&0 && \text{(by Lemma 13.1)}
\end{aligned}$$

and

$$\begin{aligned}
&-\frac{L}{m}z_\alpha \mathrm{Cov}[D_{t-m-1},\hat{\sigma}^L_{et}] \\
=&-\frac{L}{m}z_\alpha \mathrm{Cov}\left[\frac{1}{\rho}(D_{t-m}-d-\epsilon_{t-m}),\hat{\sigma}^L_{et}\right] && \text{(by (13.1))}\\
=&-\frac{L}{m}z_\alpha \frac{1}{\rho}\mathrm{Cov}[D_{t-m},\hat{\sigma}^L_{et}] && \text{(since } D, \epsilon \text{ are independent)}\\
=&0 && \text{(by Lemma 13.1).}
\end{aligned}$$

Therefore, we can ignore the $\mathrm{Cov}[\cdot]$ term in (13.11). Then using (13.10) again, we have

$$\begin{aligned}
\mathrm{Var}[Q_t] =& \left[\left(1+\frac{L}{m}\right)^2 \mathrm{Var}[D_{t-1}] + \left(\frac{L}{m}\right)^2 \mathrm{Var}[D_{t-m-1}]\right.\\
&\left. -2\left(1+\frac{L}{m}\right)\left(\frac{L}{m}\right)\mathrm{Cov}[D_{t-1},D_{t-m-1}]\right] + z_\alpha^2 \mathrm{Var}[\hat{\sigma}^L_{et}-\hat{\sigma}^L_{e,t-1}]\\
=&\left(1+\frac{2L}{m}+\frac{2L^2}{m^2}\right)\mathrm{Var}[D] - \left(\frac{2L}{m}+\frac{2L^2}{m^2}\right)\rho^m \mathrm{Var}[D]\\
&+ z_\alpha^2 \mathrm{Var}[\hat{\sigma}^L_{et}-\hat{\sigma}^L_{e,t-1}] \qquad \text{(by (13.4))}\\
=&\left[1+\left(\frac{2L}{m}+\frac{2L^2}{m^2}\right)(1-\rho^m)\right]\mathrm{Var}[D] + z_\alpha^2 \mathrm{Var}[\hat{\sigma}^L_{et}-\hat{\sigma}^L_{e,t-1}].
\end{aligned}$$

This gives us the following theorem:

Theorem 13.2

$$\frac{\mathrm{Var}[Q]}{\mathrm{Var}[D]} \geq 1 + \left(\frac{2L}{m}+\frac{2L^2}{m^2}\right)(1-\rho^m) \tag{13.13}$$

The bound is tight when $z_\alpha = 0$.

Theorem 13.2 demonstrates that demand forecasting in the presence of positive lead times is sufficient to create the BWE at a single stage. Moreover, it provides a lower bound on the percentage increase in variability. For shorthand, let B equal the lower bound on $\mathrm{Var}[Q]/\mathrm{Var}[D]$, i.e., the right-hand side of (13.13). Theorem 13.2 demonstrates that:

- As m increases, B decreases. This is intuitive since larger m means smoother forecasts, so less variability in the order sizes.

- As L increases, B increases. This is also reasonable since longer lead times make it harder to forecast demand, so the forecasts themselves, and hence the order sizes, will be more variable.

- If $\rho \geq 0$ (positively correlated demand), then as ρ increases, B decreases. The intuitive explanation is that stronger positive correlation means there is more information available to make forecasts since each demand observation also provides information about past and future demands.

- If $\rho < 0$ (negatively correlated demand), then as $|\rho|$ increases, B decreases if m is even and increases if m is odd. At first it seems surprising that the directional change in B should depend on whether m happens to be odd or even, but here is an explanation. Suppose $\rho \approx -1$, so that the demand alternates between large and small values. If m is even, then the moving average always includes the same number of large and small values, so the forecast does not change much from period to period. On the other hand, if m is odd, then the moving average itself will alternate between large and small values. Therefore, B will be smaller if m is even than if it is odd, and the difference between these two cases will be more exaggerated as $\rho \to -1$.

- If $z_\alpha = 0$, then the bound given in Theorem 13.2 is tight. In this case, no safety stock is held and stockouts occur in 50% of the periods. Simulation results given by Chen et al. (2000) suggest that even when $z_\alpha \neq 0$, the bound given by the theorem is reasonably tight.

Theorem 13.2 establishes that the BWE occurs when the demand is autocorrelated *and* the parameters are unknown. In fact, either of these conditions, by itself, is sufficient to cause the BWE. If demands are independent over time (i.e., $\rho = 0$ and the retailer knows this) but d and σ are still unknown, then Theorem 13.2 still applies and $B > 1$, so the BWE occurs. If, on the other hand, demands are still serially correlated but d, ρ, and σ are known, then the BWE occurs as well; see Problem 13.4 or Zhang (2004).

13.2.3 Rationing Game

Suppose the supply for a given product may be insufficient to meet the demand from multiple retailers and that the supplier will ration the available supply according to the fraction of demand accounted for by each retailer: If a retailer accounted for 8% of the total demand, it will receive 8% of the available supply. The BWE occurs when retailers anticipate the shortage since they have an incentive to inflate their orders to try to gain a larger share of the available supply. This behavior is called the *rationing game* because retailers play a "game" (in the game-theory sense) to try to obtain a larger allocation in the face of the supplier's rationing.

We will consider the following simple model. There are two identical retailers, each facing demand with pdf $f(\cdot)$ and cdf $F(\cdot)$ (the same distribution for both retailers). There is no inventory carryover between periods and unmet demands at the retailers are lost; therefore, we can model a single period and treat the multiperiod problem as multiple copies of the single-period one. Each unit on hand at the end of a period incurs a cost of h, and each lost sale incurs a stockout penalty of p.

Let Q^* be the optimal order quantity if the supply were infinite; that is,

$$Q^* = F^{-1}\left(\frac{p}{h+p}\right)$$

(from (4.17)). We assume that the available supply A can take on two quantities: It will equal A_1 with probability r and ∞ with probability $1-r$, with $A_1 < 2Q^*$. That is, with probability r, there will be a supply shortage, and with probability $1-r$, there will be adequate supply. (Lee et al. (1997a) consider a model with N retailers and a more general supply process, but the simpler model presented here conveys most of the same insights.)

If a retailer expects a supply shortage, it has an incentive to order more than Q^*. We will evaluate the *Nash equilibrium* solution—the order quantities chosen by the two retailers such that neither retailer, knowing the other's order quantity, would want to change its own. Put another way, a retailer's Nash equilibrium solution is the order quantity it chooses assuming it knows the other retailer's order quantity already.

Let Q_i be the order size for retailer i, $i = 1, 2$. If $A = A_1$, then retailer i will receive $A_1 Q_i/(Q_1+Q_2)$ units. For convenience, define retailer 1's allocation as

$$a(Q) = \frac{A_1 Q}{Q + Q_2}.$$

If Q_2 is fixed, retailer 1's expected cost is given by

$$\begin{aligned} g_1(Q_1) =& (1-r)\left[h\int_0^{Q_1}(Q_1-d)f(d)dd + p\int_{Q_1}^{\infty}(d-Q_1)f(d)dd\right] \\ + \quad & r\left[h\int_0^{a(Q_1)}(a(Q_1)-d)f(d)dd + p\int_{a(Q_1)}^{\infty}(d-a(Q_1))f(d)dd\right]. \end{aligned} \tag{13.14}$$

The first term of (13.14) represents the expected cost when $A = \infty$, while the second term assumes $A = A_1$. An analogous expression describes retailer 2's expected cost.

Theorem 13.3 *Both retailers choose an order quantity Q that is larger than the optimal newsvendor order quantity, Q^*.*

Proof. Retailer 1 minimizes (13.14) by setting its first derivative to 0:

$$\begin{aligned} \frac{dg_1}{dQ_1} =& r\left[h\int_0^{a(Q_1)}\frac{A_1Q_2}{(Q_1+Q_2)^2}f(d)dd + p\int_{a(Q_1)}^{\infty} -\frac{A_1Q_2}{(Q_1+Q_2)^2}f(d)dd\right] \\ & + (1-r)\left[h\int_0^{Q_1}f(d)dd + p\int_{Q_1}^{\infty}-f(d)dd\right] \quad \text{(using Leibniz's rule (C.49))} \end{aligned}$$

$$
\begin{aligned}
=& r\left[(h+p)\frac{A_1Q_2}{(Q_1+Q_2)^2}F(a(Q_1)) - p\frac{A_1Q_2}{(Q_1+Q_2)^2}\right] \\
&+ (1-r)\left[(h+p)F(Q_1) - p\right]
\end{aligned}
$$

Since retailers 1 and 2 are identical, they will make exactly the same decisions: $Q_1 = Q_2 = Q$. Then $a(Q_1) = A_1/2$. Each retailer will set

$$
r\frac{A_1}{4Q}\left[(h+p)F\left(\frac{A_1}{2}\right) - p\right] + (1-r)[(h+p)F(Q) - p] = 0.
$$

Now,

$$
F\left(\frac{A_1}{2}\right) < F\left(\frac{2Q^*}{2}\right) = F(Q^*) = \frac{p}{h+p}
$$

since $A_1 < 2Q^*$ and $F(\cdot)$ is strictly increasing. Therefore,

$$
(h+p)F\left(\frac{A_1}{2}\right) - p < (h+p)\frac{p}{h+p} - p = 0.
$$

So the optimal Q satisfies

$$
(h+p)F(Q) - p = [\text{something positive}] \tag{13.15}
$$

while the optimal Q from the newsvendor problem satisfies

$$
(h+p)F(Q) - p = 0. \tag{13.16}
$$

Since $F(\cdot)$ is strictly increasing, it takes a *larger* value of Q to satisfy (13.15) than to satisfy (13.16). ■

Therefore, in the presence of supply shortages, order quantities will be inflated. However, this, by itself, does not prove that the BWE occurs in the rationing game, since inflated order quantities do not necessarily imply inflated variances. However, Lee et al. (1997a) argue that the theorem

> ...implies the bullwhip effect when the mean demand changes over time. Retailers' equilibrium order quantity may be identical or close to the newsvendor solution for low-demand periods, while it will be larger than the newsvendor solution for high-demand periods. Hence, the variance is amplified at the retailer.

It takes some additional work to prove this claim rigorously. In fact, it can be shown that, if the mean demand changes over time as described in the quote above, then there is *no* finite Nash equilibrium in the rationing game defined by Lee et al. (1997a). That is, the retailers will keep inflating their order quantities in response to one another *ad infinitum*. However, under some minor modifications, a Nash equilibrium does exist, and its variance is greater than that of the demand, as suggested in the quote. (See Rong et al. (2017b) for these results.)

13.2.4 Order Batching

We will model the batching of orders by assuming that a given retailer will not place an order in every time period. Instead, each retailer uses a periodic-review base-stock policy with a *reorder interval* of R periods—that is, every Rth period, the retailer places an order whose size is equal to the demand seen by the retailer in the previous R periods. (See Section 4.3.4.1.) If the supplier serves several retailers, we will show that the variance of the orders seen by the supplier is larger than the variance of the orders seen by the retailers.

Suppose that there are N retailers; retailer i sees a demand of D_{it} in period t, with $D_{it} \sim N(\mu, \sigma^2)$. Demands are independent among retailers and across time periods. We consider three cases corresponding to how the retailers' orders line up with one another: random ordering, positively correlated ordering, and balanced ordering.

13.2.4.1 Random Ordering Suppose each retailer's ordering period is chosen randomly from $1, \dots, R$ with equal probability. Let X be a random variable indicating the number of orders seen by the supplier in a given time period. Since each retailer orders with probability $1/R$ in a given time period, X is a binomial random variable with parameters N and $1/R$, and

$$\begin{aligned}\mathbb{E}[X] &= \frac{N}{R} \\ \mathrm{Var}[X] &= \frac{N}{R}\left(1 - \frac{1}{R}\right).\end{aligned}$$

Let Q_t^r be the total size of the orders received by the supplier in period t. Without loss of generality, assume that retailers $1, \dots, X$ are the retailers that order in period t and retailers $X+1, \dots, N$ are the retailers that do not. Then

$$Q_t^r = \sum_{i=1}^{X} \sum_{k=t-R}^{t-1} D_{ik}.$$

(The superscript r stands for "random.") Then

$$\mathbb{E}[Q_t^r] = \mathbb{E}[\mathbb{E}[Q_t^r|X]] = \mathbb{E}[XR\mu] = N\mu,$$

where the notation $\mathbb{E}[\mathbb{E}[Q_t^r|X]]$ means we take the expectation of Q_t^r for fixed X, then take the expectation over X. Similarly,

$$\begin{aligned}\mathrm{Var}[Q_t^r] &= \mathbb{E}[\mathrm{Var}[Q_t^r|X]] + \mathrm{Var}[\mathbb{E}[Q_t^r|X]] \\ &= \mathbb{E}[XR\sigma^2] + \mathrm{Var}[XR\mu] \\ &= N\sigma^2 + R^2\mu^2\frac{N}{R}\left(1 - \frac{1}{R}\right) \\ &= N\sigma^2 + \mu^2 N(R-1) \\ &\geq N\sigma^2.\end{aligned}$$

(The first equality is a well-known identity for variance.) Therefore, the variance of orders seen by the supplier is greater than or equal to that of the demands seen by the retailers. Note that if $R = 1$ (no order batching: every retailer orders every time period), the variances are equal, as expected.

13.2.4.2 Positively Correlated Ordering We'll consider the extreme case in which all retailers order in the same period. For example, if R is 1 week, then all retailers order on Monday (say) and not on other days of the week. This is the MRP "hockey stick" taken to its extreme. The distribution function of X (the number of retailers ordering on a given day) is then

$$\mathbb{P}(X = i) = \begin{cases} 1 - 1/R, & \text{if } i = 0 \\ 1/R, & \text{if } i = N \\ 0, & \text{otherwise,} \end{cases}$$

with

$$\begin{aligned} \mathbb{E}[X] &= \frac{N}{R} \\ \mathrm{Var}[X] &= \frac{N^2}{R}\left(1 - \frac{1}{R}\right). \end{aligned}$$

Let Q_t^c be the total size of the orders received by the supplier in period t. Then

$$\mathbb{E}[Q_t^c] = \mathbb{E}[\mathbb{E}[Q_t^c|X]] = \mathbb{E}[XR\mu] = N\mu$$

and

$$\begin{aligned} \mathrm{Var}[Q_t^c] &= \mathbb{E}[XR\sigma^2] + \mathrm{Var}[XR\mu] \\ &= N\sigma^2 + R^2\mu^2\frac{N^2}{R}\left(1 - \frac{1}{R}\right) \\ &= N\sigma^2 + \mu^2 N^2(R-1) \\ &\geq N\sigma^2. \end{aligned}$$

Again, the variance of orders is greater than the variance of demands, unless $R = 1$.

13.2.4.3 Balanced Ordering Finally, suppose that the retailers' orders are evenly spread throughout the R-period reorder interval. We'll write the number of retailers N as $N = MR + k$ for integers M and k. M is like N `div` R and k is like N `mod` R. For example, if $R = 7$ (1-week reorder interval) and $N = 38$, then $M = 5$ and $k = 3$. Three days a week, six retailers order, and four days a week, five retailers order. More generally, the retailers are divided into R groups, each ordering on a different day. k of the groups have size $M + 1$ and $R - k$ of them have size M.

We get:

$$\mathbb{P}(X = i) = \begin{cases} 1 - k/R, & \text{if } i = M \\ k/R, & \text{if } i = M + 1 \\ 0, & \text{otherwise} \end{cases}$$

Then

$$\begin{aligned} \mathbb{E}[X] &= M\left(1 - \frac{k}{R}\right) + \frac{(M+1)k}{R} = \frac{N}{R} \\ \mathrm{Var}[X] &= \left(1 - \frac{k}{R}\right)M^2 + \frac{(M+1)^2 k}{R} - \left(\frac{N}{R}\right)^2 \\ &= \frac{k}{R}\left(1 - \frac{k}{R}\right). \end{aligned}$$

Let Q_t^b be the total size of the orders received by the supplier in period t. Then

$$\mathbb{E}[Q_t^b] = \mathbb{E}[\mathbb{E}[Q_t^b|X]] = \mathbb{E}[XR\mu] = N\mu$$

and

$$\begin{aligned}
\mathrm{Var}[Q_t^b] &= \mathbb{E}[XR\sigma^2] + \mathrm{Var}[XR\mu] \\
&= N\sigma^2 + R^2\mu^2\frac{k}{R}\left(1 - \frac{k}{R}\right) \\
&= N\sigma^2 + \mu^2 k(R-k) \\
&\geq N\sigma^2.
\end{aligned}$$

Once again, the variance of orders is greater than or equal to that of demands. If $k = 0$ or $k = R$, then exactly the same number of retailers place orders on each day, and the variances are equal.

We now have the following theorem.

Theorem 13.4 *Let Q_t^r, Q_t^c, and Q_t^b be random variables representing the orders received by the supplier in period t in the cases of random ordering, correlated ordering, and balanced ordering, respectively. Then:*

(a) $\mathbb{E}[Q_t^c] = \mathbb{E}[Q_t^r] = \mathbb{E}[Q_t^b] = N\mu$

(b) $\mathrm{Var}[Q_t^c] \geq \mathrm{Var}[Q_t^r] \geq \mathrm{Var}[Q_t^b] \geq N\sigma^2$

Proof. The analysis above proves (a) and the last inequality in (b). It remains to show $\mathrm{Var}[Q_t^c] \geq \mathrm{Var}[Q_t^r] \geq \mathrm{Var}[Q_t^b]$:

$$\begin{aligned}
\mathrm{Var}[Q_t^c] &= N\sigma^2 + \mu^2 N^2(R-1) \\
&\geq N\sigma^2 + \mu^2 N(R-1) \\
&= \mathrm{Var}[Q_t^r]
\end{aligned}$$

since $N \geq 1$, and

$$\begin{aligned}
\mathrm{Var}[Q_t^r] &= N\sigma^2 + \mu^2 N(R-1) \\
&\geq N\sigma^2 + \mu^2 k(R-k) \\
&= \mathrm{Var}[Q_t^b]
\end{aligned}$$

since $k(R-k) \leq N(R-1)$ for all $k = 1, \ldots, R$. ■

Therefore, the orders placed to the supplier have the same mean as those placed to the retailers, but larger variance. Moreover, correlated demand produces the largest BWE, then random, then balanced.

13.2.5 Price Speculation

We will consider a single retailer whose supplier alternates between two prices, c^L and c^H, with $c^L < c^H$. With probability r, the price will be c^L and with probability $1-r$, the price

Time Period	Starting Inventory	Demand	Price (L/H)	Order Size
1	100	77	L	77
2	100	67	H	17
3	50	82	L	132
4	100	93	L	93
etc.				

Figure 13.3 BWE spreadsheet simulation.

will be c^H. The long-run expected discounted cost, over an infinite horizon, can be written recursively as a dynamic program (DP), similar to (4.36):

$$\theta^i(x) = \min_{y \geq x}\{c^i(y-x) + g(y) + \gamma \mathbb{E}_D[r\theta^L(y-D) + (1-r)\theta^H(y-D)]\}, \quad (13.17)$$

where $i \in \{L, H\}$ and, as usual, $g(y)$ is as given by (4.37). Note that we have two recursive functions, one for each cost level. The recursion (13.17) differs from (4.36) in two respects. First, the expected future cost contains an expectation over the cost level i. Second, this is an infinite-horizon recursion, so $\theta^i(x)$ does not have a time-period index, and the definition of $\theta^i(x)$ depends on itself. Dynamic programming has tools to deal with this sort of recursion, which we will not explore here. Suffice it to say that the optimal inventory policy in this case can be shown to be a modified base-stock policy: When the price is c^L, order enough to bring the inventory position to S^L, and when the price is c^H, order enough to bring the inventory position to S^H. If the inventory level is greater than the applicable base-stock level in a given period, returns are not allowed; instead, the retailer orders 0. It is clear that $S^H \leq S^L$, but finding the optimal S^H and S^L can be difficult. We omit the details here. The net result is the following theorem:

Theorem 13.5 $\text{Var}[Q] > \text{Var}[D]$

Therefore, price fluctuations produce the BWE.

You can get a feel for how this works by building a spreadsheet simulation model. For example, Figure 13.3 shows the first few rows of a spreadsheet that has columns for starting inventory, demand (we used $N(80, 100)$ to generate demand), price (low or high; we used $r = 0.7$), and order size (we used $S^L = 100$, $S^H = 50$ to compute these, but these are not the optimal base-stock levels). The results of the simulation are displayed graphically in Figure 13.4. The orders clearly display a larger variance than the demands.

13.3 REDUCING THE BULLWHIP EFFECT

A number of strategies have been proposed for addressing the four causes of the BWE. We discuss some of these next.

13.3.1 Demand Signal Processing

The analysis given above suggests that the BWE is amplified as we move upstream in the supply chain since stage i uses stage $(i+1)$'s orders as though they were demands, when

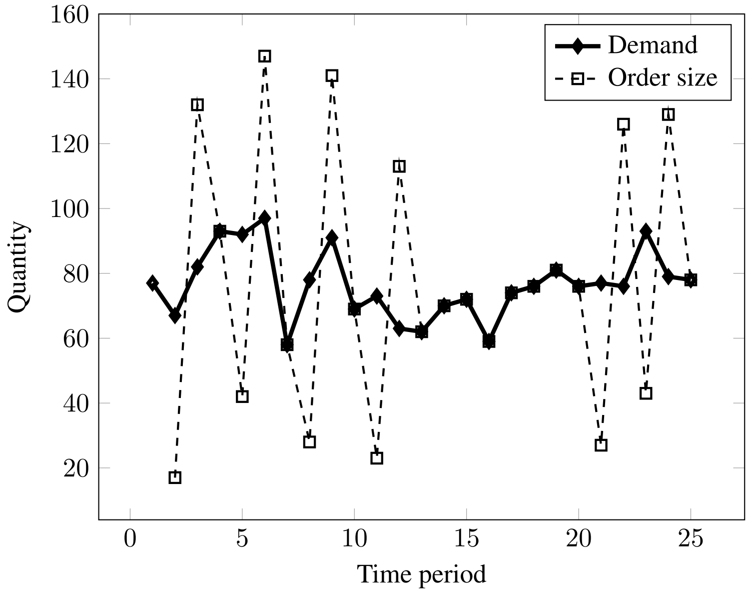


Figure 13.4 BWE caused by price fluctuations.

in fact they are more variable than demands. This can be mitigated by sharing *point-of-sale* (POS) demand information with upstream members of the supply chain. That is, when the retailer places an order with the wholesaler, it relays not only the order size but also the size of the most recent demands. The proliferation of bar code scanners at checkout lines makes this technologically easy, but retailers are often reluctant to give demand data, which they treat as proprietary, to their suppliers. In addition, even if upstream stages see this "sell-through" data, they may each use different forecasting techniques or inventory policies, and this will exacerbate the BWE as well. We will analyze the effect of sell-through data on the BWE in Section 13.4.

Vendor-managed inventory (VMI) is a distribution strategy whereby the vendor (say, Coca-Cola) manages the inventory at the retailer (say, Walmart). The Coca-Cola company sets up the Coke displays at Walmart and, more importantly, monitors the inventory level and replenishes as necessary. In many cases, Coke actually owns the merchandise until it is sold—Walmart only takes ownership of the product for a split second as it's being scanned at the checkout line. Walmart benefits because Coke pays some of the costs of holding and managing the inventory. Coke benefits because it can keep tighter control over the displays of its products at stores, and also because its distribution is more efficient when it, not its customers, decides when to replenish the stock at each store. Moreover, since Coke gets to see actual sales data, the BWE is reduced.

As we saw earlier, longer lead times make the BWE worse. Therefore, one strategy for reducing the BWE is to shorten lead times. There are various ways to accomplish this, though it is often easier said than done.

13.3.2 Rationing Game

Rather than rationing according to order sizes in the current period, the supplier could allocate the available supply based on each retailer's orders in the *previous* period, or based on market share or some other mechanism that's independent of this period's orders. That eliminates the incentive to over-order during shortages. Alternatively, the supplier could restrict each retailer's orders to be no more than a certain percentage (say 10%) larger than its order in the previous period, or charge a small "reservation payment" for each item ordered, whether or not it is received. Finally, the supplier can avoid the rationing game to a certain extent by sharing supply information with downstream members (note the symmetry with the demand signal processing case), allowing the retailers to use actual data instead of conjecture when making ordering decisions.

13.3.3 Order Batching

Recall from the EOQ model (Section 3.2) that as the fixed order cost increases, so does the order size. The batching of orders, then, can be reduced by reducing the fixed order cost. Nowadays, most communication uses *electronic data interchange* (EDI), in which communication is performed electronically instead of on paper. This reduces the cost in both time and money of placing each order. Another innovation that reduces the setup cost of each order is *third-party logistics* (3PL) providers, which allow smaller companies to attain larger economies of scale by taking advantage of the 3PL's size. For example, if a firm wants to ship a single package to a customer, it doesn't have to contract for a full truck—it can just use UPS, one of the world's largest 3PLs. Since UPS has lots of packages going all over the world, it attains huge economies of scale and passes some of these savings to its customers.

Suppliers can also encourage less batching by offering retailers volume discounts based on their total order, not based on orders for individual products. For example, P&G used to give bulk discounts if retailers ordered an entire truckload of one product (say, Pampers); now they give the same discounts even if the truck carries a variety of P&G products. This allows retailers to order Pampers more frequently (possibly with every order) as opposed to only ordering Pampers when they need a full truckload.

If batching is unavoidable, suppliers can force the orders to be balanced over time by assigning each retailer a specific period during which it may place orders. For example, one retailer might have to place orders only on Tuesdays, while another may place orders on Thursdays. This strategy will reduce, but not avoid, the BWE, as we saw in Theorem 13.4.

13.3.4 Price Speculation

One way to avoid the variability introduced by price fluctuations is simply to keep prices fixed. Although this seems obvious, it has introduced a shift in the pricing schemes of many major manufacturers such as P&G, Kraft, and Pillsbury. The strategy is called *everyday low pricing* (EDLP), and the basic idea is that prices stay at a constant low rate: there are no sales or promotions. EDLP is widely used upstream in the supply chain, but it is also increasingly used for retail sales. You may have seen stores that advertise "everyday low prices" and assumed it is merely a marketing ploy, without realizing the substantial benefit the retailer may be gaining by reducing the BWE.

In some cases, price fluctuations are unavoidable or desirable, and a natural consequence is that retailers will buy more when the price is low. The supplier can still reduce the BWE, however, by proposing contracts in which the retailer agrees to buy a large quantity of goods at a discount but to spread the receipt of the goods over time. The manufacturer can plan production more efficiently, but the retailer can continue to buy when prices are low.

13.4 CENTRALIZING DEMAND INFORMATION

In Section 13.3, we suggested that sharing POS demand information with upstream supply chain members reduces the BWE: Instead of seeing the retailer's orders, which are already more variable than the demands, the supplier sees the actual demands and uses these to make its own ordering decisions. But can this strategy eliminate the BWE entirely? If not, how much can it reduce the BWE?

In this section, we will analyze the impact of demand sharing on the BWE using the model introduced in Section 13.2.2, extending the analysis now to multiple stages as pictured in Figure 13.2. We will consider a *centralized* system in which each stage sees the actual customer demands; we will then compare this system to a *decentralized* system in which demand information is not shared and each stage sees only the orders placed by its immediate downstream neighbor.

The lead time for goods being transported from stage i to stage $i+1$ is given by L_{i+1}. Each stage uses a moving average forecast with m observations. The moving average is used to compute estimates of the lead time demand mean, μ_t^L, and the standard deviation of the forecast error of lead-time demand, σ_{et}^L, which are in turn used to compute the base-stock levels.

13.4.1 Centralized System

In the centralized system, demand information is available to all stages of the supply chain. There is no "information lead time"—all stages see customer demands at exactly the same moment, when the demands arrive. Stage i can build its moving average forecast using actual customer demands. Its estimates of μ_t^L and σ_{et}^L will be as given in (13.7) and (13.8), and it will use these to compute base-stock levels as in (13.9).

Conceptually, there is no difference between (a) goods being shipped from i to $i+1$ to ... to N to the customer, with a total lead time of $L_{i+1} + L_{i+2} + \cdots + L_N$, and (b) goods being shipped directly from i to the customer with the same lead time. Therefore, we can think of stage i as serving the end customer demand directly with a transportation lead time of $L_{i+1} + L_{i+2} + \cdots + L_N$. Using the same logic as in Section 13.2, we get the following theorem, which quantifies the increase in variability between the customer demands and the orders placed by a given stage:

Theorem 13.6 *In a centralized serial supply chain, the variance of the orders placed by stage i, denoted Q_i, satisfies*

$$\frac{\mathrm{Var}[Q_i]}{\mathrm{Var}[D]} \geq 1 + \left(\frac{2\sum_{j=i+1}^{N} L_j}{m} + \frac{2\left(\sum_{j=i+1}^{N} L_j\right)^2}{m^2} \right)(1-\rho^m)$$

for all $i = 1, \ldots, N$.

Thus, even if (1) demand information is visible to all supply chain members, (2) all supply chain members use the same forecasting technique, and (3) all supply chain members use the same inventory policy, *the bullwhip effect still exists*. Sharing demand information does not eliminate the BWE. But does it reduce it? We will answer this question in the next section by comparing this system to one in which demand information is not shared.

13.4.2 Decentralized System

Consider the same system as in the previous section except that demand information is not shared: Each stage only sees the orders placed by its downstream stage. For simplicity, we will assume that $\rho = 0$ (demands are uncorrelated across time). We will also assume that $z_\alpha = 0$ (a 50% service level is acceptable), which means no safety stock is held. (Firms sometimes use inventory policies of this form, inflating L_i artificially to provide a buffer against uncertainty. For example, the firm might increase L_i by 7 days, requiring 7 extra *days of supply* of inventory to be on hand at any given time. Firms generally refer to this inflated lead time as *safety lead time*, but we can think of safety lead time as essentially an alternate method of setting safety stock.)

The "demands" seen by stage i are really the orders placed by stage $i+1$. The variance of these orders is at least $1 + 2L_{i+1}/m + 2L_{i+1}^2/m^2$ times the variance of the orders received by stage $i+1$, by Theorem 13.2. By following this logic through to stage N, we get the following theorem:

Theorem 13.7 *In a decentralized serial supply chain with $\rho = 0$ and $z_\alpha = 0$, the variance of the orders placed by stage i, denoted Q_i, satisfies*

$$\frac{\mathrm{Var}[Q_i]}{\mathrm{Var}[D]} \geq \prod_{j=i+1}^{N} \left(1 + \frac{2L_j}{m} + \frac{2L_j^2}{m^2}\right)$$

for all $i = 1, \ldots, N$.

Therefore, the increase in variability is additive *in the centralized system but* multiplicative *in the decentralized system.* Sharing demand information can significantly reduce the BWE. Although our analysis of the decentralized system assumed $\rho = z_\alpha = 0$, the qualitative result (additive vs. multiplicative variance increase) still holds in the more general case, though the math is uglier.

To get a sense of the difference in magnitude between the bounds provided by Theorems 13.6 and 13.7, consider the case in which $N = 4$, $L_i = 2$ for all i, and $\rho = z_\alpha = 0$. Then the right-hand sides of the inequalities are given in Table 13.1. Note how much larger the bounds are for the decentralized system, especially as we move upstream in the supply chain.

CASE STUDY 13.1 Reducing the Bullwhip Effect at Philips Electronics

High-tech products typically have very volatile demand, long lead times, and short product lifecycles. Moreover, the manufacture and assembly of many key components are often outsourced to third parties. As a result, high-tech supply chains tend to be fragmented, with several independent firms, each optimizing their own objective functions. This can easily lead to the BWE, if upstream firms have poor visibility

Table 13.1 Bounds on variability increase: Decentralized vs. centralized.

i	Decentralized	Centralized
1	12.7	7.2
2	6.7	5.0
3	3.6	3.2
4	1.9	1.9

into downstream demands and if upstream shortages lead to rationing-game behavior downstream.

In 2000, Philips Electronics began a major project to reduce the bullwhip effect in its supply chain, as described by de Kok et al. (2005). At the time, Philips was one of the world's largest electronics companies, with sales of over $30 billion and over 150,000 employees around the world. Philips Semiconductors (PS) and Philips Optical Storage (POS) are[1] subsidiaries of Philips Electronics (highlighting the fragmented nature of the supply chain); PS manufactures semiconductors at 20 sites worldwide, and POS produces DVD drives and other optical storage devices. POS is PS's customer, and the BWE was evident in the ordering patterns between these two firms and others in the supply chain. The two firms decided to implement a collaborative planning approach across the supply chain to improve visibility and coordination, as well as to optimize inventory and material-flow decisions. They partnered with researchers from the Technische Universiteit Eindhoven (Technical University of Eindhoven, the Netherlands) and a consulting firm to develop the approach.

Figure 13.5 depicts a schematic of the supply chain for DVD drives. Semiconductor wafers are fabricated by PS and then assembled into integrated circuits (ICs). The ICs are then sent to POS, which uses them to make optical pickup units (OPUs), and to various third-party subcontractors, which use them to make flex units and printed circuit boards (PCBs). POS then assembles the OPUs, flex units, and PCBs (the latter two of which are sent from subcontractors) into DVD drives. From there, the drives are sent to both brick-and-mortar and online retailers. The average processing times at each stage (in weeks) are written below the stages. Inventory buffers exist at each stage. It is worth noting that in addition to the long processing time, wafer fabrication is subject to significant yield uncertainty and limited capacity.

Prior to the BWE-reduction project, communication among the stages in the supply chain was poor, with multiple stages making decisions independently from the decisions other stages were making. This lack of coordination led to long information lead-times, with downstream demand changes taking 6 weeks or longer to affect upstream decision-making. It also led to information distortion, due to lack of visibility, and to rationing gaming, due to upstream shortages. As a result, each stage tended to hold large quantities of inventory to buffer against both supply and demand uncertainty, which in turn led to significant obsolescence risk. Even so, both PS and POS often missed their delivery deadlines.

[1] We will use the present tense, though PS and POS both now take different forms, after mergers, spinoffs, and the like.

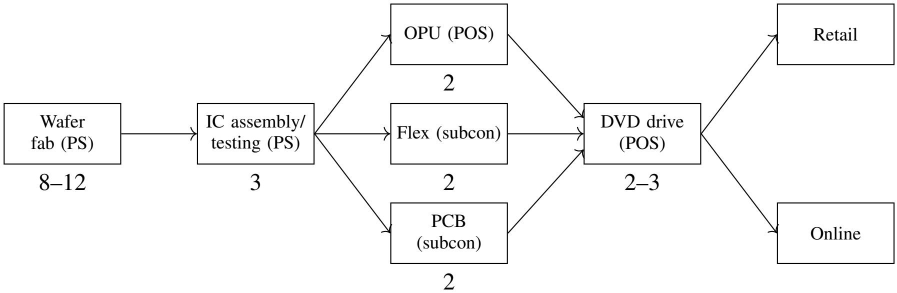


Figure 13.5 Philips Optical Storage's DVD supply chain network.

The project team chose to use *safety lead times* to buffer against the uncertainty. As noted in Section 13.4.2, a safety lead time is an extra quantity of time added to the regular lead time, which has the effect of creating extra inventory in the system via formulas like (4.46) or (13.6). They set larger safety lead times at the start of a product's lifecycle to protect against unforeseen demand spikes and smaller lead times later on, as the demand became more predictable. Alternately, the safety lead times could have been set using multiechelon inventory theory; for example, de Kok and Visschers (1999) and de Kok and Fransoo (2003) extend the concepts from Chapter 6 to more general supply chain structures.

Let i be a product (stage) in the supply chain, and let F_i be the set of end-products in which product i is used. For the sake of simplicity, we will assume that each unit of product $k \in F_i$ uses exactly one unit of product i. Let L^*_{ik} and ST^*_{ik} be the total lead time and safety lead time, respectively, along the path from i to k (including both i and k). Finally, let $\hat{D}_{it}$ be the forecast of the demand for product i in period t. Then the base-stock level for product i is

$$S_i = \sum_{k \in F_i} \sum_{t=1}^{L^*_{ik}+ST^*_{ik}+1} \hat{D}_{it}, \tag{13.18}$$

where the right-hand side of (13.18) equals the forecast of the lead-time demand for product i. This is analogous to (4.46), except that (1) the demand for the upstream product is calculated by aggregating over all downstream products that it is used in, (2) we are allowing the demand to be nonstationary, and (3) rather than setting the safety stock level using an estimate of the demand standard deviation, we are using a safety lead time, which inflates the base-stock level when it gets multiplied by the demand. However, we can still determine the safety stock: As always, it equals the expected ending inventory level, i.e., the base-stock level minus the lead time demand:

$$SS_i = S_i - \sum_{k \in F_i} \sum_{t=1}^{L^*_{ik}+1} \hat{D}_{it}.$$

The base-stock levels become inputs to a mathematical optimization problem that decides the order or production quantities in each period at each stage; for details, see de Kok et al. (2005).

The model, algorithm, and collaborative planning process were implemented in 2001. In the years that followed, stakeholders at PS and POS reported better communication, shorter information lead times, and greater supply chain visibility. Between 2001 and 2003, the companies reported significant reductions in finished-goods inventory levels and obsolescence at the end of product lifecycles, with a combined benefit of $5 million. They also reported increased flexibility in responding to upswings in demand, leading to a 1.5% increase in profits. Additional benefits included improved delivery reliability, better supply–demand balance, and a reduction in the bullwhip effect. As de Kok et al. (2005) conclude, "All this has led to a provably synchronized supply chain from die to DVD and, for Philips, a solution that finally knocks the bullwhip effect on the head."

PROBLEMS

13.1 (Stochastic Price Simulation) Suppose that the price of the raw material for a given product is stochastic, as in Section 13.2.5. The price equals $c^L = 3$ with probability 0.8 and equals $c^H = 8$ with probability 0.2. Demands in each period are $N(100, 20^2)$. On-hand inventory at the end of each period incurs a holding cost of $h = 1$ per unit. Unmet demands are backordered with a stockout cost of $p = 20$ per unit.

The firm uses two base-stock levels, S^L and S^H, ordering up to the appropriate level in each period based on the current price. For now, assume $S^L = 200$ and $S^H = 100$.

a) Simulate this system using spreadsheet software for 1000 periods. Build a table like the one in Figure 13.3 listing the time period, starting inventory, demand, price, order size, and any other columns you find useful. Also indicate the total cost in each period, including holding, stockout, and order costs.

b) Using a spreadsheet-based nonlinear optimization package, determine the values of S^L and S^H that minimize the average cost per period for the random sample you have generated. Report the optimal S^L and S^H and the resulting average cost per period. Include the first few rows of your spreadsheet in your report.

c) Calculate $\mathrm{Var}[Q]$ and $\mathrm{Var}[D]$ for your simulation and compare them to verify that the BWE occurs.

d) Produce a chart like the one in Figure 13.4 plotting the demands and orders across the time horizon.

13.2 (Batching Simulation) In the one-warehouse, multiple-retailer (OWMR) system pictured in Figure 13.6, all three retailers, and the warehouse, handle a single product. Demands at the retailers are normally distributed with means and variances as given in Table 13.2, which also lists h, p, and K at each retailer. (As usual, h is the holding cost per item per period, p is the backorder cost per item per period, and K is the fixed cost per order placed to the warehouse.) Since $K > 0$, it's optimal for the retailers to follow an (s, S) policy rather than a base-stock policy.

a) Compute near-optimal values for s and S for each retailer using the power approximation from Section 4.4.4.

b) Simulate this system using a spreadsheet or other software package using the (s, S) values you found in part (a). Simulate at least 1000 periods, with a warm-up interval of 100 periods. Report the standard deviation of the total demands

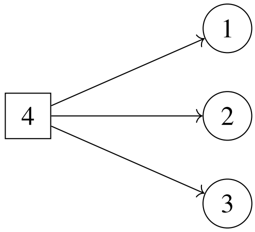


Figure 13.6 One-warehouse, multiple-retailer system for Problem 13.2.

Table 13.2 Data for Problem 13.2.

Retailer	μ	σ^2	h	p	K
1	50	8^2	0.6	7	100
2	100	22^2	0.4	8	100
3	40	3^2	0.9	4	100

seen by the retailers and the standard deviation of the total demands (retailer orders) seen by the warehouse. Using these values, verify that the BWE occurs in this system.

c) In a short paragraph, explain why the retailers' (s, S) policies cause the BWE.

13.3 (Lead-Time Demand Under Autocorrelation) For the demand process in Section 13.2.2, prove that D_t^L is normally distributed with mean

$$\mu_t^L = \frac{d}{1-\rho}\left(L - \rho\frac{1-\rho^L}{1-\rho}\right) + D_{t-1}\rho\frac{1-\rho^L}{1-\rho}$$

and standard deviation

$$\sigma_t^L = \frac{\sigma}{1-\rho}\sqrt{L - 2\rho\frac{1-\rho^L}{1-\rho} + \rho^2\frac{1-\rho^{2L}}{1-\rho^2}}.$$

(Note that σ_t^L is independent of t.)

13.4 (BWE Occurs Even If Demand Parameters Are Known) Suppose that we know d, ρ, and σ in Section 13.2.2. Using the results in Problem 13.3, prove that

$$\frac{\text{Var}[Q]}{\text{Var}[D]} = 1 + \frac{2(1-\rho^{L+1})(\rho-\rho^{L+1})}{1-\rho}.$$

(Since this is greater than 1, the BWE occurs even if the parameters are known and therefore no forecasting is required.)

13.5 (Proving BWE in the Beer Game) In the "stationary beer game" (Chen and Samroengraja 2000), a serial supply chain faces normally distributed demands at the downstream node, and each stage orders from its supplier in each period. The optimal inventory policy at each stage is a base-stock policy: The size of the order a stage places in a given period is equal to the size of the order received by the stage in that period. If each player followed this policy, there would be no BWE since the variance of outgoing orders would be the same as that of incoming orders. The beer game is designed to illustrate the irrational behavior of supply chain managers, who tend to over-react to *perceived* trends in demand (even when no actual trend is present) by ordering more than necessary when demands are high and less than necessary when demands are low. In this problem, you will model this over-reaction mathematically and prove that it causes the bullwhip effect.

Consider a retailer who faces demand $D_t \sim N(\mu, \sigma^2)$ in period t. Demands are independent across time periods. The retailer, acting irrationally, over-orders by $\theta \geq 0$ units *for each consecutive period* in which the demand was higher than μ, including the current period. Similarly, it under-orders by θ units for each consecutive period in which the demand was lower than μ, where $\theta \geq 0$ is a constant. That is, although the optimal policy is to set the order size as $Q_t = D_t$, the retailer actually uses

$$Q_t = D_t + \theta X_t^+ - \theta X_t^-,$$

where X_t^+ is the number of consecutive periods (including t) in which the demand was greater than μ and X_t^- is the number of consecutive periods (including t) in which the demand was less than μ.

a) Prove that $\mathbb{E}[Q] = \mathbb{E}[D]$ (the retailer's mean order size is equal to the mean demand).

b) Prove that

$$\frac{\mathrm{Var}[Q]}{\mathrm{Var}[D]} \geq 1 + \frac{6\theta^2}{\mathrm{Var}[D]}.$$

Hint 1: What probability distribution describes X_t^+ and X_t^-?
Hint 2: Remember that $\mathrm{Cov}[X, Y] = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]$.

13.6 (Beer Game Simulation) In the beer game, players act irrationally (i.e., not following the optimal inventory policy). One model of this irrational behavior is by Sterman (1989), who suggests that the order quantity placed by stage i ($i = 1, \ldots, N$) in period t of the beer game can be described by the following model:

$$Q_t^i = \max\left\{0, \hat{Q}_{t+1}^{i+1} + \alpha^i(a^i - IL_t^i) + \beta^i(b^i - (IP_t^i - IL_t^i))\right\}, \qquad (13.19)$$

where

- a^i, b^i, α^i, and β^i are constants for stage i, described in more detail below
- IL_t^i = inventory level (on-hand inventory minus backorders) at stage i after step 2 in the sequence of events given below (i.e., after observing its demand but before placing its order) in period t
- IP_t^i = inventory position (on-hand inventory plus on-order inventory minus backorders) at stage i after step 2 in period t
- Q_t^i = order quantity placed by stage i in step 3 in period t; if $i = N + 1$, then Q_t^i represents demand from the external customer
- $\hat{Q}_t^i$ = forecast of order quantity that will be placed by stage i in period t; this forecast is calculated by stage $i - 1$ after step 2 in period $t - 1$ using exponential smoothing:

$$\hat{Q}_t^i = \eta Q_{t-1}^i + (1 - \eta)\hat{Q}_{t-1}^i,$$

where η is the smoothing factor, $0 \le \eta \le 1$.

The constants a^i and b^i represent target values for the inventory level (IL^i) and on-order inventory ($IP^i - IL^i$), respectively, for stage i. The constants α and β are adjustment parameters controlling the change in order quantity when the actual inventory level and the on-order inventory, respectively, deviate from the desired targets.

The sequence of events at stage i in each period of the beer game is as follows:

1. The shipment from stage $i-1$ shipped two periods ago arrives at stage i. (If $i = 1$, stage $i-1$ refers to the manufacturing process at the farthest upstream stage.)

2. The order placed by stage $i+1$ in the current period is observed by stage i. (If $i = N$, stage $i+1$ refers to the external customer.) The order from stage $i+1$, plus any backorders that stage $i+1$ is waiting for, is satisfied using the current on-hand inventory, and excess demands are backordered.

3. Stage i determines its order quantity and places its order to stage $i-1$.

4. Holding and/or stockout costs are incurred.

a) Using MATLAB, Excel, or any other software package of your choice, simulate the beer game under the assumption that all players use (13.19) to set their order quantities. Assume that $N = 4$. Model the demand from the external customer in each period as an iid random variable distributed as $N(50, 10^2)$. Set $\eta = 0.1$, $a^i = 10$, $b^i = 100$, and $\alpha^i = \beta^i = 0.5$ for all i. Initialize the system by assuming that $\hat{Q}_1^i = 50$ and $IL_0^i = 0$ for all i, and that there are 50 units of in-transit inventory due to arrive in each of period 1 and period 2 for each i. (That is, assume that each stage has 1 period's worth of inventory on-hand and in each in-transit slot.) Report the magnitude of the BWE at each stage, defined as $\text{Var}[Q^i]/\text{Var}[Q^{i+1}]$.

b) Conduct a numerical experiment to evaluate how the BWE changes as the players' order behavior changes. At a minimum, use your experiment to answer the following questions:

- Does the BWE get more or less severe when stages increase the weight α^i they place on the on-hand inventory level?
- Does the BWE get more or less severe when stages increase the weight β^i they place on the on-order inventory?
- Does the BWE get more or less severe when stages increase their target levels a^i and b^i?
- Does the BWE get more or less severe when stages increase the smoothing constant η?
- Suppose stage i uses (13.19) to set order quantities but all other players are more rational, setting $\alpha^i = \beta^i = 0$. Which produces more severe BWE—having the "irrational" player upstream or downstream in the supply chain?

Support your analysis with numerical results, preferably in graph (chart) form.

CHAPTER 14

SUPPLY CHAIN CONTRACTS

14.1 INTRODUCTION

Supply chains are typically composed of multiple players, each with competing goals. For example, the newsvendor wants to pay a small wholesale cost per unit to the supplier, but the supplier wants a large wholesale cost. If each player acts selfishly, the resulting solution is generally suboptimal for the supply chain as a whole—the total profit earned by the supply chain is smaller than if the players could somehow bring their actions in line with one another. (By "selfishly" we don't mean they're behaving meanly or inappropriately—just that each player naturally acts in his or her own best interest, making decisions to maximize his or her own profit.)

In the past few decades, a great deal of research has studied *contracting mechanisms* for achieving supply chain *coordination*—for enticing each player to act in such a way that the total supply chain profit is maximized. The basic idea is that the players agree on a certain contract that specifies a payment, called a *transfer payment*, made from one party to another. The size of the transfer payment can be determined in any number of ways (and identifying these ways are the focus of much of the research). Many are quite intuitive: For example, the retailer might pay a wholesale price to the supplier but receive a credit for unsold merchandise at the end of the period (like the newsvendor's wholesale price and salvage value). If these mechanisms are designed correctly, then even when each player acts in his or her own best interest, the supply chain profit is maximized.

Fundamentals of Supply Chain Theory, Second Edition. Lawrence V. Snyder and Zuo-Jun Max Shen.

Companion website: www.wiley.com/go/Snyder/SupplyChainTheory

In this chapter, we return to the newsvendor problem, now considering the newsvendor's supplier as an active player in the game. We will show that under the assumptions studied previously, the newsvendor (whom we'll now refer to as the *retailer*) does not order enough inventory to maximize the total supply chain profit. We will then introduce a few contract types that coordinate the newsvendor model. The material in this chapter originates from Pasternack (1985) and other sources cited below, as well as Cachon (2003), who reviews many of the basic ideas of supply chain coordination. We first review some important concepts from game theory.

14.2 INTRODUCTION TO GAME THEORY

The literature on supply chain coordination draws heavily from game theory. There are many textbooks on game theory, e.g., Osborne (2003); see also the review of game theory as it applies to supply chain analysis by Cachon and Netessine (2004). We will not cover game theory formally here, but it is worth introducing a few terms. A *game* consists of two or more *players* (we will assume exactly two). Each player may choose from a set of *strategies*, and a choice of strategies (one for each player) is called an *outcome*. For each outcome, there is a *payoff* to each player.

For example, if there are two players (A and B), each with two strategies (1 and 2), the payoffs might be as given in Table 14.1. Player A's payoff is the first number in the pair, player B's is the second number. If player A chooses strategy 1 and player B chooses strategy 2, the payoff is -4 to player A (a loss) and 2 to player B.

Table 14.1 Payoffs for a sample game.

		Player B: 1	Player B: 2
Player A	1	$(1, 1)$	$(-4, 2)$
	2	$(2, -4)$	$(-2, -2)$

Note that there is no randomness in this game. The term "outcome" refers to the deterministic result of choices that the players make, not to the result of some random experiment. In the games we will consider, there is also some randomness that determines the payoffs, in which case the "outcome" represents the expected payoffs to the players.

An outcome is called *Pareto optimal* if there is no other outcome in which both players have higher payoffs, or in which one player has a higher payoff and the other player has the same payoff. For example, in Table 14.1, the outcome in which both players choose strategy 1 is Pareto optimal, since one player can't be made better off without making the other worse off. Pareto optimal outcomes are considered to be "fair" in some sense.

A *Nash equilibrium* is an outcome such that neither player can change strategies unilaterally and improve his or her own payoff. (Nash equilibrium is named after the mathematician and economist John Nash.) If the players act selfishly, the game will move to a Nash equilibrium. There is one Nash equilibrium in the game depicted above: Each player chooses strategy 2. (You should verify that this is the only Nash equilibrium in the game.) However, this outcome is not Pareto optimal, since both players would be better off if they each chose strategy 1.

(The game in Table 14.1 is an example of a *prisoner's dilemma*. In a prisoner's dilemma, the Nash equilibrium is different from the Pareto optimal solution, so the players will always find themselves at an undesirable solution (the Nash equilibrium) even though a mutually better solution (the Pareto optimal solution) is available.)

Now suppose that the players entered into the following simplistic contract: At the end of the game, the players will equally split any profit or loss. The resulting payoff structure is given in Table 14.2.

Table 14.2 Payoffs after implementing a contract.

		Player B	
		1	2
Player A	1	$(1, 1)$	$(-1, -1)$
	2	$(-1, -1)$	$(-2, -2)$

Now the Nash equilibrium is for both players to choose strategy 1 (neither player has any incentive to change strategies), and this strategy is also Pareto optimal. This is the outcome the players would have preferred in the original game, but acting individually they would never have arrived at that outcome. By introducing a simple contract, the players choose the best solution, even when they act in their own interest.

Notice that the contract does not *force* any player to choose a strategy other than the one that maximizes his or her outcome. That is, it does not force the players to choose the outcome $(1, 1)$. It simply restructures the payoffs so that the players *want* to choose that outcome.

In the supply chain context, we will see that the Nash equilibrium outcome, to which the players would gravitate if acting in their own interest, is generally not Pareto optimal—there are other outcomes that would improve the payoff to both players. The goal of supply chain coordination is to change the structure of the payoffs so that the Nash equilibrium is also Pareto optimal. One important question will be whether, in the resulting Nash equilibrium, both players earn more than they did without the contract. (If not, one party may refuse to enter into the contract.) The goal of supply chain contracts is not to force one player to earn a smaller piece of the pie so that the other player can earn a bigger piece. Rather, it's to make the pie bigger so that both players can get bigger pieces than they had before.

The games presented in Tables 14.1 and 14.2 are called *static games* because the two players choose their strategies simultaneously (though a player may alter his or her strategy in response to the other player's strategy). Supply chain contracts, however, are a different type of game, namely, a *Stackelberg game*, in which one player chooses a strategy first and then the other player chooses one. (Stackelberg games are also known as *leader–follower games*.) The models presented below are based on the newsvendor model, and in these models, the supplier is the leader, setting the parameters of the contract, and the newsvendor is the follower, setting the order quantity.

14.3 NOTATION

As in the classical newsvendor model, we consider a single-period model with stochastic demand. Let D be the demand during the period, with mean μ, pdf f, and cdf F. The retail price (i.e., revenue per unit sold by the retailer) is r per unit. The supplier's production cost is c_s per unit and the retailer's cost is c_r per unit. Note that c_r does *not* get paid to the supplier—it represents the cost of processing, shipping, marketing, etc., at the retailer. It is incurred when the unit is procured from the supplier, not when it is sold. We assume $c_s + c_r < r$ (otherwise the system cannot make any profit).

Unsatisfied demands are lost (since this is a one-period model), incurring a stockout penalty of p_r at the retailer and p_s at the supplier. These costs reflect the loss-of-goodwill that the parties incur; they do not include the lost profit resulting from a lost sale. This is because the profit is already explicitly calculated in this model, so including lost profit in p_r and p_s would double-count this penalty. (Similarly, see the explicit formulation of the newsvendor problem in Section 4.3.2.4.) For convenience, we let $c = c_s + c_r$ and $p = p_s + p_r$. Each unsold unit at the retailer at the end of the season can be salvaged for a salvage value of v per unit, with $v < c_r$. The retailer's order size is denoted Q.

The notation is summarized in Table 14.3.

Table 14.3 Contracting notation summary.

D	$\sim f, F$
μ	$= \mathbb{E}[D]$
r	selling price
c_s, c_r	supplier's, retailer's per-unit cost
c	$= c_s + c_r$
p_s, p_r	supplier's, retailer's loss-of-goodwill cost
p	$= p_s + p_r$
v	salvage value
Q	retailer's order size

14.4 PRELIMINARY ANALYSIS

The following sequence of events occurs in the game:

1. The supplier chooses her[1] contract parameters. (Each contract type has its own parameters.)

2. The retailer chooses his order quantity Q and places his order to the supplier; the order arrives immediately.

3. Demand occurs; as much as possible is satisfied from inventory, and the rest is lost.

4. Costs are assessed and transfer payments are made between the players.

The transfer payment depends on the type of contract, several of which will be explored below. Note that we are assuming that the supplier offers the contract to the retailer—that the supplier is the powerful player in the market. This is not necessarily the case, and other models have explored the newsvendor problem when the retailer is the powerful player.

[1]We'll use the common convention in the contracting literature that the supplier is female and the retailer is male.

Our first goal is to formulate the supplier's and retailer's expected cost as functions of Q. To that end, let $S(Q)$ be the expected sales as a function of Q:

$$\begin{aligned} S(Q) &= \mathbb{E}[\min\{Q, D\}] \\ &= \mathbb{E}[Q - (Q - D)^+] \\ &= Q - \bar{n}(Q), \end{aligned} \tag{14.1}$$

where $\bar{n}(Q)$ is the complementary loss function. The second equality follows from (C.5), while the third follows from (C.13). Then letting $\bar{F}(Q) = 1 - F(Q)$,

$$S'(Q) = \bar{F}(Q) \tag{14.2}$$

by (C.16). Let $I(Q)$ be the expected inventory on hand at the end of the period:

$$I(Q) = \mathbb{E}[(Q - D)^+] = \bar{n}(Q) = Q - S(Q). \tag{14.3}$$

Let $L(Q)$ be the expected lost sales:

$$\begin{aligned} L(Q) &= \mathbb{E}[(D - Q)^+] = n(Q) \\ &= \mu - Q + \bar{n}(Q) \\ &= \mu - S(Q). \end{aligned} \tag{14.4}$$

(The third equality follows from (C.14).) Finally, let T be the expected transfer payment (whose size is yet to be determined).

The retailer's expected profit function is then

$$\begin{aligned} \pi_r(Q) &= rS(Q) + vI(Q) - p_r L(Q) - c_r Q - T \\ &= rS(Q) + v(Q - S(Q)) - p_r(\mu - S(Q)) - c_r Q - T \\ &= (r - v + p_r)S(Q) - (c_r - v)Q - p_r\mu - T. \end{aligned} \tag{14.5}$$

$\pi_r(Q)$ is basically just a newsvendor cost function, written in a very different way—maximizing profit rather than minimizing cost (but the two are mathematically equivalent) and writing the expectations using the functions $S(\cdot)$, $I(\cdot)$, and $L(\cdot)$. The supplier's expected profit function is

$$\begin{aligned} \pi_s(Q) &= -c_s Q - p_s L(Q) + T \\ &= -c_s Q - p_s(\mu - S(Q)) + T \\ &= p_s S(Q) - c_s Q - p_s\mu + T. \end{aligned} \tag{14.6}$$

The supply chain's total expected profit function is therefore

$$\Pi(Q) = \pi_r(Q) + \pi_s(Q) = (r - v + p)S(Q) - (c - v)Q - p\mu. \tag{14.7}$$

Let's find the order quantity Q^0 that maximizes the total supply chain profit.

$$\begin{aligned} & \Pi'(Q^0) = 0 \\ \iff & (r - v + p)S'(Q^0) - (c - v) = 0 \\ \iff & S'(Q^0) = \bar{F}(Q^0) = \frac{c - v}{r - v + p}. \end{aligned} \tag{14.8}$$

Q^0 is a maximizer, not a minimizer, because

$$\Pi''(Q) = -(r - v + p)f(Q) < 0,$$

so Π is concave.

Equation (14.8) agrees with our previous results from the newsvendor model. In particular, if we think of the supply chain as a whole acting as the newsvendor, with per-unit cost c, sales price r, penalty cost p, and salvage value v, then the newsvendor has costs $h' = c - v$ and $p' = r - c + p$, and $h' + p' = r - v + p$. From (4.17), the optimal newsvendor order quantity satisfies

$$F(Q) = \frac{p'}{h' + p'}$$

or

$$\bar{F}(Q) = 1 - \frac{p'}{p' + h'} = \frac{c - v}{r - v + p}.$$

The question now is, does the retailer choose Q^0 as his order quantity? And, is this also the order quantity that the supplier prefers? That is, if Q_r^* and Q_s^* maximize (14.5) and (14.6) (respectively), then does $Q_r^* = Q_s^* = Q^0$?

The supply chain is considered coordinated if $Q_r^* = Q_s^* = Q^0$. A contract type is said to coordinate the supply chain if there exist contract parameters such that $Q_r^* = Q_s^* = Q^0$ *and* the players each earn positive profit. If the optimal order quantities coincide but one player earns a negative profit, the player's willingness to enter into the contract depends on a number of factors, such as the player's profit under the status quo (which could, after all, be even more negative), the other business relationships the players may jointly have, the players' relative levels of power, and so on. We ignore these rather messy issues and focus below on determining which contract types are guaranteed to have parameters such that the supply chain is coordinated and the players both earn positive profits.

14.5 THE WHOLESALE PRICE CONTRACT

The simplest possible contract is the *wholesale price contract*, in which the retailer pays the supplier a given cost w per unit ordered. This is identical to settings we've discussed previously, in which the retailer pays a per-unit purchase cost that goes to the supplier, except now the purchase cost is the supplier's decision variable. For a given wholesale cost w, the transfer payment is given by

$$T_w(Q, w) = wQ.$$

The subscript w identifies the type of contract, while the arguments specify the two decision variables—order quantity and wholesale cost—one per player.

The retailer's and supplier's expected profits are both functions of w and Q:

$$\begin{aligned} \pi_r(Q, w) &= (r - v + p_r)S(Q) - (c_r - v)Q - p_r\mu - wQ \\ &= (r - v + p_r)S(Q) - (w + c_r - v)Q - p_r\mu \qquad (14.9) \\ \pi_s(Q, w) &= p_s S(Q) + (w - c_s)Q - p_s\mu \qquad (14.10) \end{aligned}$$

The supply chain is coordinated if there exists a value of w such that $Q_r^* = Q_s^* = Q^0$, where Q_r^*, Q_s^*, and Q^0 are the order quantities that maximize π_r, π_s, and Π, respectively.

It is straightforward to show that π_r and π_s are both concave functions of Q (assuming that w is fixed). Therefore, Q_r^* and Q_s^* satisfy:

$$\left.\frac{\partial \pi_r(Q,w)}{\partial Q}\right|_{Q=Q_r^*} = (r - v + p_r)S'(Q_r^*) - (w + c_r - v) = 0 \tag{14.11}$$

$$\left.\frac{\partial \pi_s(Q,w)}{\partial Q}\right|_{Q=Q_s^*} = p_s S'(Q_s^*) + (w - c_s) = 0 \tag{14.12}$$

The next theorem demonstrates that there exists a value of w such that $Q_r^* = Q_s^* = Q^0$. However, for this value of w, the supplier earns a negative expected profit.

Theorem 14.1 $Q_r^* = Q_s^* = Q^0$ *if and only if*

$$w = c_s - \frac{c - v}{r - v + p} p_s. \tag{14.13}$$

Moreover, the supplier earns a negative expected profit under this wholesale price.

Proof. Suppose (14.13) holds. Then by (14.11),

$$\begin{aligned} S'(Q_r^*) &= \frac{w + c_r - v}{r - v + p_r} \\ &= \frac{\left(c_s - \frac{c-v}{r-v+p} p_s\right) + c_r - v}{r - v + p_r} \\ &= \frac{(c - v)\left(1 - \frac{p_s}{r-v+p}\right)}{r - v + p_r} \\ &= \frac{c - v}{r - v + p} \\ &= S'(Q^0) \end{aligned}$$

by (14.8). Since $S'(Q)$ is strictly decreasing and continuous, this implies $Q_r^* = Q^0$. Similarly, by (14.12),

$$\begin{aligned} S'(Q_s^*) &= \frac{c_s - w}{p_s} \\ &= \frac{c_s - \left(c_s - \frac{c-v}{r-v+p} p_s\right)}{p_s} \\ &= \frac{c - v}{r - v + p} \\ &= S'(Q^0). \end{aligned}$$

Therefore, $Q_r^* = Q_s^* = Q^0$. However, since $v < c_r \leq c < r$ by assumption, the coefficient of p_s in (14.13) is negative, which means that $w < c_s$. Therefore, the retailer earns a negative expected profit. ■

From (14.12) one can show (see Problem 14.1) that if $w < c_s - p_s$, then π_s is strictly decreasing in Q; if $c_s - p_s < w < c_s$, then π_s is first increasing and then decreasing; and if $w > c_s$, then π_s is strictly increasing. Thus, for sufficiently large w, $Q_s^* = \infty$ since the supplier earns a positive margin on items sold to the retailer, and she pays no penalty for overage at the retailer. For moderate values of w ($c_s - p_s < w < c_s$), Q_s^* is finite: Although the supplier earns a negative margin on each unit sold to the retailer, she still prefers a nonzero order quantity since small order quantities cause stockouts, for which the supplier incurs a goodwill cost. Her expected profit is still negative, but Q_s^* minimizes the losses. Finally, for small values of w, the supplier's margin is so negative that it more than offsets the goodwill cost, and she sets $Q_s^* = 0$.

The phenomenon evident in Theorem 14.1 is known as *double marginalization* (Spengler 1950): When both players add their own margin (markup) to their costs, the supply chain is not coordinated since the players ignore the total supply chain profit when making their individual decisions. If, on the other hand, the retailer has a positive margin but the supplier has a negative one, the supply chain is coordinated. However, the supplier clearly would not enter into this arrangement, so the wholesale price contract is not considered to be a coordinating one. Nevertheless, there are still several interesting things to say about it.

Suppose that $w > c_s$, so that the supplier earns positive profit but the supply chain is not coordinated. We first examine the retailer's and supplier's optimization problems and then discuss how close the wholesale price contract comes to coordinating the supply chain.

Theorem 14.2 *Under the wholesale price contract, if $w > c_s$, then $Q_r^* < Q^0$.*

Proof. Omitted; see Problem 14.6. ■

Theorem 14.2 says that, assuming the supplier earns a positive profit, the retailer will under-order. This happens because the retailer is absorbing all of the risk of overage, but only part of the risk of underage (since the supplier pays a stockout penalty). Therefore, the retailer orders less than the supplier (and the supply chain as a whole) wants him to. In the contracting mechanisms discussed in later sections, the supplier absorbs some of the risk of overage, thus giving the retailer the flexibility to increase his order quantity. If the contract parameters are set correctly, he'll increase it so that it equals Q^0.

Now let's turn our attention to the supplier's optimization problem. For given w, (14.12) determines the supplier's optimal order quantity. But what if the supplier could choose whatever w she wants? In order to choose w, she must anticipate the Q that the retailer will choose for each value of w. Put another way, the supplier can entice the retailer to choose whatever Q she wishes by selecting the unique wholesale price, call it $w(Q)$, that makes Q optimal for the retailer. In particular,

$$w(Q) = (r - v + p_r)\bar{F}(Q) - (c_r - v) \tag{14.14}$$

(from (14.11)).

Which Q does the supplier want the retailer to choose? The supplier's profit function is now a function of Q and the corresponding $w(Q)$:

$$\pi_s(Q, w(Q)) = p_s S(Q) + (w(Q) - c_s)Q - p_s\mu \tag{14.15}$$

The supplier wishes to maximize this function. The question is, does it have a unique maximum? If the function is strictly concave, it does, but in general $\pi_s(Q, w(Q))$ is *not* concave. But it is usually close enough. Before we explain further, we need to introduce a new property that is important in contract analysis: Demand distributions for which $Qf(Q)/\bar{F}(Q)$ is increasing are called *increasing generalized failure rate* (IGFR) distributions (Lariviere and Porteus 2001, Lariviere 2006, Banciu and Mirchandani 2013). Many common distributions are IGFR, including normal, exponential, and gamma. Only a few families of distributions are *not* IGFR; these include Gumbel and generalized logistic (Paul 2005).

In the next theorem, we show that if the demand distribution is IGFR, then $\pi_s(Q, w(Q))$ is *unimodal*—that is, there is a value Q^* such that the derivative of $\pi_s(Q, w(Q))$ is positive to the left of Q^* and negative to the right. This implies that the function has a unique maximum. Unimodality (sometimes called *quasiconcavity*) is similar to strict concavity but is weaker: Every strictly concave function is unimodal, but not every unimodal function is concave.

Theorem 14.3 *If the demand distribution $f(\cdot)$ is IGFR, then $\pi_s(Q, w(Q))$ is unimodal and therefore has a unique maximum.*

Proof. The derivative of $\pi_s(Q, w(Q))$ is given by

$$\begin{aligned}
\pi_s'(Q, w(Q)) =& p_s S'(Q) + w(Q) - c_s + w'(Q)Q \\
=& p_s \bar{F}(Q) + (r - v + p_r)\bar{F}(Q) - (c_r - v) - c_s \\
& - (r - v + p_r) f(Q) Q \\
=& (r - v + p_r)\bar{F}(Q)\left(1 + \frac{p_s}{r - v + p_r} - \frac{Qf(Q)}{\bar{F}(Q)}\right) - (c - v). \qquad (14.16)
\end{aligned}$$

In (14.16),

$$1 + \frac{p_s}{r - v + p_r} - \frac{Qf(Q)}{\bar{F}(Q)} \qquad (14.17)$$

is positive for $Q \approx 0$ (since $\bar{F}(Q) \approx 1$) and negative for $Q \to \infty$ (see Problem 14.16). Since the derivative is continuous, there must be some value Q^* at which the derivative equals 0. We need to prove that there is only one such value. To do so, it suffices to show that $\pi_s(Q, w(Q)) > 0$ for all $Q < Q^*$ and $\pi_s(Q, w(Q)) < 0$ for all $Q > Q^*$.

We know that $\bar{F}(Q)$ is decreasing and positive, and that (14.17) is decreasing (since $f(\cdot)$ is IGFR) and positive when $Q = Q^*$ (otherwise, the right-hand side of (14.16) could not equal 0). Therefore, as Q decreases from Q^*, both $\bar{F}(Q)$ and (14.17) increase; so $\pi_s'(Q, w(Q)) > 0$ for $Q < Q^*$. And as Q increases from Q^*, both $\bar{F}(Q)$ and (14.17) decrease, so $\pi_s'(Q, w(Q))$ decreases *if* (14.17) is positive. On the other hand, (14.17) could be negative; but in this case $\pi_s'(Q, w(Q)) < 0$ since $\bar{F}(Q) > 0$. Therefore, $\pi_s(Q, w(Q))$ is unimodal. ■

In what follows, we will assume that the demand distribution is IGFR. Hence $\pi_s(Q, w(Q))$ is unimodal, and there is a unique order quantity that maximizes the supplier's profit. (Note that this is the order quantity the supplier would choose if she can also choose w. This is not the same as Q_s^* as defined in (14.12), which is the supplier's optimal quantity for fixed w.) Of course, the supplier does not set this order quantity directly; she sets w to $w(Q_s^*)$ and waits for the retailer to set the order quantity to Q_s^*.

For contracts such as the wholesale price contract that do not coordinate the supply chain, we'd like to know how close they come. This is measured by the *efficiency* of the contract: the proportion of the optimal supply chain profit attained by the Nash equilibrium order quantity, or $\Pi(Q_s^*)/\Pi(Q^0)$. The greater the efficiency, the closer the contract is to achieving coordination. Another important measure is the *supplier's profit share*: the percentage of the total profit captured by the supplier, or $\pi_s(Q_s^*, w(Q_s^*))/\Pi(Q_s^*)$. Both players would like the efficiency to be high, but only the supplier would like the supplier's profit share to be high. Experimental tests using the power distribution show the efficiency to be around 75% and the supplier's profit share to be in the range of 55–80% (Cachon 2003). Perakis and Roels (2007) examine the efficiency of the wholesale price contract in a variety of settings.

It is worth noting that the wholesale price contract is considered to be noncoordinating because there is no value of w that (1) makes $Q_r^* = Q_s^* = Q^0$ and (2) guarantees both players positive profits. In contrast, the contract types we discuss in the next sections are considered to be coordinating because there always exist some values of the contract parameters for which both (1) and (2) hold, even though not all parameter values do the trick.

☐ **EXAMPLE 14.1**

Matilda's Market sells bagels that are made by Jeffrey's Bakery. Bagels sell for $\$1$. Each bagel that the market buys from the bakery costs the bakery \$0.50 to make and costs the market \$0.25 in processing costs. Daily demand for bagels is distributed as $N(100, 20^2)$. Unmet demands incur a loss-of-goodwill cost of \$0.20 for each party. Unsold bagels must be thrown out at the end of each day, with no salvage value. Currently, Jeffrey's Bakery charges Matilda's Market a wholesale price of \$0.60 per bagel. What is Matilda's Market's optimal order quantity? What is each company's profit, and what is the profit of the supply chain as a whole?

We have $r = 1$, $c_s = 0.5$, $c_r = 0.25$, $p_r = p_s = 0.2$, $v = 0$, and $w = 0.6$. From (14.2) and (14.11), Q_r^* satisfies

$$F(Q_r^*) = 1 - \frac{0.85}{1.2}$$
$$\Longrightarrow \; Q_r^* = 89.03.$$

From (14.1), (C.22), and (C.32) one can calculate $S(89.03) = 85.3648$. From (14.9), (14.10), and (14.7),

$$\pi_r(89.03, 0.6) = 1.2S(89.03) - 0.85 \cdot 89.03 - 0.2 \cdot 100 = 6.7627$$
$$\pi_s(89.03, 0.6) = 0.2S(89.03) + 0.1 \cdot 89.03 - 0.2 \cdot 100 = 5.9759$$
$$\Pi(89.03) = 6.7627 + 5.9759 = 12.7386.$$

How does this compare to the optimal total profit? From (14.8),

$$F(Q^0) = 1 - \frac{0.75}{1.4}$$
$$\Longrightarrow \; Q^0 = 98.21.$$

We have $S(Q^0) = 91.0927$, so from (14.7), the total profit is

$$\Pi(Q^0) = 1.4S(98.21) - 0.75 \cdot 98.21 - 0.4 \cdot 100 = 13.8744.$$

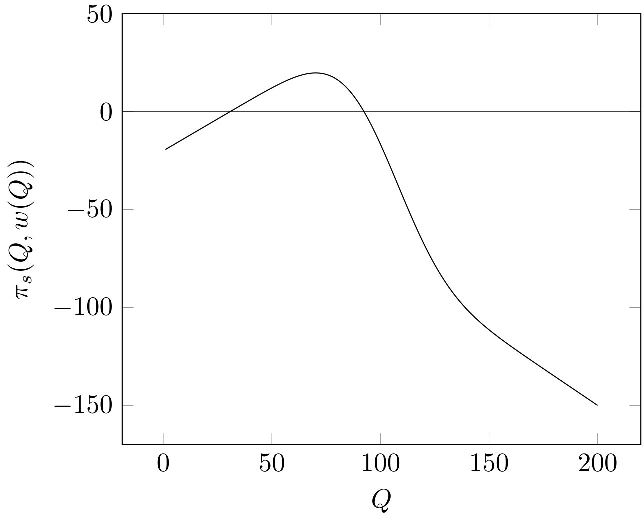


Figure 14.1 Supplier's expected profit function $\pi_s(Q, w(Q))$ for Example 14.1.

Therefore, the total profit under this wholesale price contract is less than the optimal supply chain profit.

Now suppose that Jeffrey's Bakery can choose whichever w it wants. What w will it choose, what Q will Matilda's Market choose as a result, and what will be the resulting profits?

The supplier's profit function, $\pi_s(Q, w(Q))$, is plotted in Figure 14.1. Note that it is unimodal, as promised by Theorem 14.3, but is not concave. The expected profit is only positive for a relatively narrow range of Q values.

From (14.16), the Q that maximizes Jeffrey's Bakery's profit is the Q that satisfies

$$1.2\bar{F}(Q)\left(1 + \frac{0.2}{1.2} - \frac{Qf(Q)}{\bar{F}(Q)}\right) - 0.75 = 0.$$

The reader can verify that this equation is satisfied (within rounding error) by $Q = 70.23$. Therefore, Jeffery's Bakery will choose

$$w(Q) = 1.2\bar{F}(70.23) - 0.25 = 0.8680$$

and Matilda's Market will choose $Q = 70.23$. Note that $S(70.23) = 69.6283$. The profits will be

$$\begin{aligned}
\pi_r(70.23, 0.8680) &= 1.2S(70.23) - 1.1180 \cdot 70.23 - 0.2 \cdot 100 = -14.9652 \\
\pi_s(70.23, 0.8680) &= 0.2S(70.23) + 0.3680 \cdot 70.23 - 0.2 \cdot 100 = 19.7723 \\
\Pi(70.23) &= -14.9652 + 19.7723 = 4.8072.
\end{aligned}$$

Jeffrey's Bakery may like this arrangement, but it is clearly bad for Matilda's Market, and it is also suboptimal for the supply chain as a whole.

$\square$

14.6 THE BUYBACK CONTRACT

We now examine a contract type that does coordinate the supply chain. In the *buyback contract* (Pasternack 1985), the supplier charges the retailer w per unit purchased, but pays the retailer b for every unit of unsold inventory at the end of the period:

$$T_b(Q, w, b) = wQ - bI(Q) = bS(Q) + (w - b)Q.$$

We assume

$$0 \leq b \leq r - v + p_r, \tag{14.18}$$

otherwise it is better for the retailer *not* to sell an on-hand item to satisfy a demand (thereby earning $b + v$ but paying p_r) than to sell it (earning r). We also assume

$$b \leq w + c_r - v, \tag{14.19}$$

otherwise, the retailer incurs no overage risk since his revenue for salvaging an item $(b + v)$ is more than what he paid for it $(w + c_r)$.

The name "buyback" is a little misleading, because usually the retailer does not physically return the products, he just receives a credit from the supplier. The supplier is offering to share some of the risk of overage with the retailer in exchange for higher supply chain profits.

Many suppliers offer buyback credits as a way to prevent the unsold goods from being sold at a steep discount. For example, high-fashion clothing makers don't want to see their products on the bargain rack at Marshall's at the end of the season, so they give high-end retailers a credit to prevent them from unloading unsold merchandise to discounters.

Letting $T = T_b(Q, w, b)$ in (14.5), the retailer's profit function becomes

$$\begin{aligned}\pi_r(Q, w, b) &= (r - v + p_r)S(Q) - (c_r - v)Q - p_r\mu - [bS(Q) + (w - b)Q] \\ &= (r - v + p_r - b)S(Q) - (c_r - v + w - b)Q - p_r\mu. \end{aligned} \tag{14.20}$$

The Q that maximizes the retailer's profit satisfies

$$\begin{aligned}\frac{\partial \pi_r(Q, w, b)}{\partial Q} &= (r - v + p_r - b)\bar{F}(Q) - (c_r - v + w - b) = 0 \\ \iff \bar{F}(Q) &= \frac{c_r - v + w - b}{r - v + p_r - b}. \end{aligned} \tag{14.21}$$

Now, the supplier can, of course, choose any values for w and b (subject to (14.18) and (14.19)). However, it turns out that for a given b, the "correct" value of w (i.e., the one that will coordinate the supply chain) is given by

$$w(b) = b + c_s - (c - v)\frac{b + p_s}{r - v + p}. \tag{14.22}$$

(It should not yet be obvious how we get this value of $w(b)$.) Note that $w(b)$ is increasing in b (since $r > c$). Therefore, in exchange for receiving a more generous buyback credit, the retailer must pay a higher wholesale price.

If w and b satisfy (14.18) and (14.22), then they also satisfy (14.19). (See Problem 14.7.) Therefore, we can ignore (14.19) and assume only that the feasible region for b is $[0, r - v + p_r]$.

Theorem 14.4 establishes that, if the parameters are set appropriately, then the optimal order quantities coincide. Moreover, by Theorem 14.5 below, there exists a b such that both parties earn positive profit. Therefore, the buyback contract coordinates the supply chain.

Theorem 14.4 *Under the buyback contract, for any b satisfying* (14.18)*, if $w(b)$ is set according to* (14.22)*, then $Q_r^* = Q_s^* = Q^0$.*

We'll present two proofs of this theorem. The first is more straightforward than the second, but the second is quite elegant and can also be applied in more general settings.

Proof #1. Substituting $w(b)$ into (14.21), we get

$$\begin{aligned}\bar{F}(Q) &= \frac{c_r - v + \left[b + c_s - (c-v)\frac{b+p_s}{r-v+p}\right] - b}{r - v + p_r - b} \\ &= \frac{(c-v)\left[1 - \frac{b+p_s}{r-v+p}\right]}{r - v + p_r - b} \\ &= \frac{(c-v)\frac{r-v+p_r-b}{r-v+p}}{r - v + p_r - b} \\ &= \frac{c-v}{r-v+p}.\end{aligned}$$

This is the same Q that maximizes the total supply chain expected profit (see (14.8)), so $Q_r^* = Q^0$. Therefore, from the retailer's perspective, the supply chain is coordinated. It remains to show that the supplier also prefers this same Q.

Letting $T = T_b(Q, w, b)$ in (14.6), the supplier's profit function is

$$\begin{aligned}\pi_s(Q, w, b) &= p_s S(Q) - c_s Q - p_s\mu + [bS(Q) + (w-b)Q] \\ &= (p_s + b)S(Q) - (c_s - w + b)Q - p_s\mu. \qquad (14.23)\end{aligned}$$

The Q that maximizes π_s satisfies

$$\begin{aligned}\frac{\partial \pi_s(Q, w, b)}{\partial Q} &= (p_s + b)\bar{F}(Q) - (c_s - w + b) = 0 \\ \iff \bar{F}(Q) &= \frac{c_s - w + b}{p_s + b}. \qquad (14.24)\end{aligned}$$

Letting $w = w(b)$ as defined in (14.22), we get

$$\begin{aligned}\bar{F}(Q) &= \frac{c_s - \left[b + c_s - (c-v)\frac{b+p_s}{r-v+p}\right] + b}{p_s + b} \\ &= \frac{(c-v)\frac{b+p_s}{r-v+p}}{p_s + b} \\ &= \frac{c-v}{r-v+p}.\end{aligned}$$

Therefore, $Q_s^* = Q^0$. ∎

Before introducing the second proof, we can answer the question of how to determine $w(b)$ (if it wasn't already given by (14.22)): It's the only value of w that makes the conditions $\partial\pi_r/\partial Q = 0$ and $\partial\pi_s/\partial Q = 0$ both reduce to $\bar{F}(Q) = (c - v)/(r - v + p)$. The value of $w(b)$ can be "backed out" from these conditions.

In general, we can use the approach from Proof #1—setting $\partial\pi_r/\partial Q$ and $\partial\pi_s/\partial Q$ to 0 and showing that $Q_r^* = Q_s^* = Q^0$—to prove that a given contracting mechanism coordinates the supply chain. However, there's another elegant way to accomplish this, and this approach is taken by Proof #2.

Proof #2. Let

$$\lambda = \frac{r - v + p_r - b}{r - v + p}. \tag{14.25}$$

Then

$$\begin{aligned}
\lambda &= 1 - \frac{b + p_s}{r - v + p} \\
&= \frac{c - v - (c - v)\frac{b+p_s}{r-v+p}}{c - v} \\
&= \frac{b + c_s - (c - v)\frac{b+p_s}{r-v+p} - b + c_r - v}{c - v} \\
&= \frac{w(b) - b + c_r - v}{c - v}
\end{aligned} \tag{14.26}$$

In other words, λ is equal to the fractions in both (14.25) and (14.26). (This is a result of our definition of $w(b)$.) Since $p_r \le p$ and $b \ge 0$, $\lambda \le 1$. Also, since $b \le r - v + p_r$ (by (14.18)), $\lambda \ge 0$.

From (14.20), the retailer's profit under a buyback contract is

$$\begin{aligned}
\pi_r(Q, w, b) &= (r - v + p_r - b)S(Q) - (w - b + c_r - v)Q - p_r\mu \\
&= \lambda(r - v + p)S(Q) - \lambda(c - v)Q - p_r\mu \\
&= \lambda\Pi(Q) + \mu(\lambda p - p_r),
\end{aligned} \tag{14.27}$$

where $\Pi(Q)$ is the total supply chain profit as defined in (14.7). Since $\lambda \ge 0$, the same Q minimizes (14.7) and (14.27), so $Q_r^* = Q^0$. The same argument applies to the supplier since

$$\pi_s(Q, w, b) = \Pi(Q) - \pi_r(Q, w, b) = (1 - \lambda)\Pi(Q) - \mu(\lambda p - p_r) \tag{14.28}$$

and $\lambda \le 1$. ∎

As λ increases, the retailer's profit increases and the supplier's profit decreases, so in a sense λ represents the division of profit between the players. One would like to know whether *any* division is possible—that is, is there some value of λ such that the supplier captures all of the profit and another value such that the retailer captures all of the profit? (Keep in mind that λ is not a parameter of the contract—the supplier does not actually choose λ. But by choosing b, the supplier automatically chooses λ given (14.25).) If so, then there is also a value that gives any desired mix. As the next theorem demonstrates, this is indeed possible.

Theorem 14.5 *If $w(b)$ is set according to* (14.22)*, then the retailer's [supplier's] profit is decreasing [increasing] in $b \in [0, r - v + p_r]$. Moreover, let*

$$b_1 = r - v + p_r - (r - v + p)\frac{\Pi(Q^0) + \mu p_r}{\Pi(Q^0) + \mu p} \tag{14.29}$$

$$b_2 = r - v + p_r - (r - v + p)\frac{\mu p_r}{\Pi(Q^0) + \mu p}. \tag{14.30}$$

Then $0 < b_1 < b_2 < r - v + p_r$, and:

1. *If $0 \le b < b_1$, then the supplier earns negative profit (and the retailer earns more than $\Pi(Q^0)$).*
2. *If $b = b_1$, then the retailer earns the entire supply chain profit.*
3. *If $b_1 < b < b_2$, then the profits are shared by the players.*
4. *If $b = b_2$, then the supplier earns the entire supply chain profit.*
5. *If $b_2 < b \le r - v + p_r$, then the retailer earns negative profit (and the supplier earns more than $\Pi(Q^0)$).*

Proof. We first prove $0 < b_1 < b_2 < r - v + p_r$. The second and third inequalities follow immediately from the fact that $v < r$. To prove $0 < b_1$, first suppose that the demand is deterministic; then the supply chain earns a profit of $\mu(r - c)$ since there is no overage or underage. This provides an upper bound on the maximum possible expected profit under stochastic demand, i.e., $\Pi(Q^0) \le \mu(r - c)$. Since $v < c_r \le c$, we have $\Pi(Q^0) < \mu(r - v)$, and therefore

$$\frac{\mu(r - v) + \mu p_r}{\mu(r - v) + \mu p} > \frac{\Pi(Q^0) + \mu p_r}{\Pi(Q^0) + \mu p}.$$

Thus, $b_1 > 0$.

It remains to prove items 1–5. By (14.27) and (14.28), the retailer's profit $\pi_r(Q^0, w(b), b)$ is an increasing function of λ and the supplier's profit $\pi_s(Q^0, w(b), b)$ is a decreasing function of λ. By (14.25), λ is a decreasing function of b. Therefore, the retailer's [supplier's] profit is a decreasing [increasing] function of b. Since the sum of their profits is fixed (to $\Pi(Q^0)$), to prove the theorem it suffices to prove items 2 and 4.

If $b = b_1$, then

$$\lambda = \frac{r - v + p_r - b_1}{r - v + p} = \frac{\Pi(Q^0) + \mu p_r}{\Pi(Q^0) + \mu p} \tag{14.31}$$

$$\implies \Pi(Q^0) = \lambda \Pi(Q^0) + \mu(\lambda p - p_r) = \pi_r(Q^0, w(b), b)$$

by (14.27). Similarly, if $b = b_2$, then

$$\lambda = \frac{\mu p_r}{\Pi(Q^0) + \mu p} \tag{14.32}$$

$$\implies \Pi(Q^0) = (1 - \lambda)\Pi(Q^0) - \mu(\lambda p - p_r) = \pi_s(Q^0, w(b), b)$$

by (14.28). ■

At first it may seem surprising that the supplier's profit is increasing in b, since b is a payment made to the retailer. However, $w(b)$ is increasing in b, and increases in the buyback credits paid to the retailer are more than offset by increased revenue from the wholesale price.

One consequence of Theorem 14.5 is that, for any noncoordinating contract, there exist b and $w(b)$ such that neither player has a lower profit under the buyback contract, and at least one player has a strictly higher profit. Therefore, the supplier can always choose b and $w(b)$ such that both players prefer the buyback contract to the status quo if the supply chain is not currently coordinated.

Which value of b *will* she choose? We can't solve this as an optimization problem as we did for the wholesale price contract because the supplier's profit is an increasing function of b. Left to her own devices, she would choose a large b that gives the retailer negative profit. Instead, the choice of b is the result of some sort of negotiation process that reflects the relative power of the two players as well as other factors, which we ignore. The contract types discussed in the following sections are similar in this regard.

□ **EXAMPLE 14.2**

Return to Example 14.1 and suppose that Jeffrey's Bakery offers a buyback contract to Matilda's Market with $b = 0.6$. What w will coordinate the supply chain, and what are the resulting profits?

From (14.22),

$$w(0.6) = 0.6 + 0.5 - 0.75 \cdot \frac{0.6 + 0.2}{1 - 0 + 0.4} = 0.6714.$$

Since the contract coordinates, both players will choose $Q = Q^0 = 98.21$. From (14.20) and (14.23),

$$\begin{aligned}\pi_r(98.21, 0.6714, 0.6) &= 0.6S(98.21) - 0.3214 \cdot 98.21 - 0.2 \cdot 100 = 3.0890\\ \pi_s(98.21, 0.6714, 0.6) &= 0.8S(98.21) - 0.4286 \cdot 98.21 - 0.2 \cdot 100 = 10.7854.\end{aligned}$$

The two profits sum to 13.8744, confirming that the supply chain is coordinated.

Figure 14.2 plots w, $\pi_r(Q^0, w(b), b)$, $\pi_s(Q^0, w(b), b)$, and $\Pi(Q^0)$ as a function of b. Note that when $b = 0.3197$, the retailer (Matilda's Market) earns all of the profit and when $b = 0.6803$, the supplier (Jeffrey's Bakery) earns all of the profit. The reader can confirm that these values equal b_1 and b_2 from Theorem 14.5. The figure therefore confirms the behavior described in the theorem. □

14.7 THE REVENUE SHARING CONTRACT

In the *revenue sharing contract* (Cachon and Lariviere 2005), the supplier charges the retailer a wholesale price of w per unit *and* the retailer gives the supplier a percentage of his revenue. All revenue is shared, including both sales revenue and salvage value. Let ϕ be the fraction of revenue the retailer keeps and $1 - \phi$ the fraction he gives to the supplier. The transfer payment is then

$$\begin{aligned}T_r(Q, w, \phi) &= wQ + (1 - \phi)rS(Q) + (1 - \phi)vI(Q)\\ &= (w + (1 - \phi)v)Q + (1 - \phi)(r - v)S(Q). \end{aligned} \qquad (14.33)$$

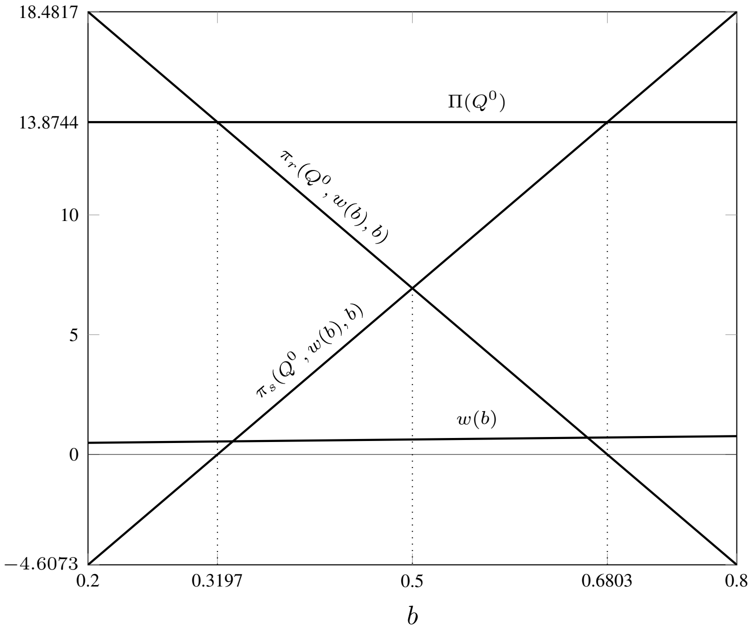


Figure 14.2 Wholesale price and profits as a function of buyback credit.

Again, we magically determine the "correct" value of one contract parameter (w) for a given value of the other (ϕ):

$$w(\phi) = -c_r + \phi v + (c - v)\frac{\phi(r - v) + p_r}{r - v + p}. \tag{14.34}$$

If we define $w(\phi)$ in this way, then the supply chain is coordinated. The next theorem demonstrates this; its proof uses a method similar to Proof #2 in Section 14.6, but it could also be proven using a method similar to Proof #1 (see Problem 14.13).

Theorem 14.6 *Under the revenue sharing contract, for any* $0 \le \phi \le 1$*, if* $w(\phi)$ *is set according to* (14.34)*, then* $Q_r^* = Q_s^* = Q^0$*, i.e., the supply chain is coordinated.*

Proof. The retailer's profit function is

$$\begin{aligned}\pi_r(Q, w, \phi) =& (r - v + p_r)S(Q) - (c_r - v)Q - p_r\mu \\ & - [(w + (1 - \phi)v)Q + (1 - \phi)(r - v)S(Q)] \\ =& (\phi(r - v) + p_r)S(Q) - (w + c_r - \phi v)Q - p_r\mu. \end{aligned} \tag{14.35}$$

Let

$$\lambda = \frac{\phi(r - v) + p_r}{r - v + p} \le 1; \tag{14.36}$$

then also (by virtue of the relationship between w and ϕ),

$$\lambda = \frac{w + c_r - \phi v}{c - v} \ge 0. \tag{14.37}$$

(Recall that $v < c_r$.) Then

$$\pi_r(Q, w, \phi) = \lambda \Pi(Q) + \mu(\lambda p - p_r). \tag{14.38}$$

Since $\lambda \geq 0$, $Q_r^* = Q^0$, and the revenue sharing contract coordinates the supply chain from the retailer's perspective. Since the total supply chain profit is the sum of the retailer's and the supplier's, the supplier's profit is

$$\pi_s(Q, w, \phi) = \Pi(Q) - \pi_r(Q, w, \phi) = (1 - \lambda)\Pi(Q) - \mu(\lambda p - p_r). \tag{14.39}$$

Therefore, $Q_s^* = Q^0$ since $\lambda \leq 1$. ■

The retailer's profit is increasing in λ and the supplier's profit is decreasing in λ. Since λ is increasing in ϕ, the retailer's profit increases and the supplier's profit decreases as the retailer's revenue fraction ϕ increases. One can prove a theorem that is analogous to Theorem 14.5 demonstrating that any allocation of profits is possible under the revenue sharing contract. In particular, the retailer earns the entire profit if

$$\lambda = \frac{\Pi(Q^0) + \mu p_r}{\Pi(Q^0) + \mu p} \tag{14.40}$$

and the supplier earns the entire profit if

$$\lambda = \frac{\mu p_r}{\Pi(Q^0) + \mu p}. \tag{14.41}$$

(Note the similarity to (14.31) and (14.32).)

Revenue sharing and buyback contracts are actually quite similar. For the sake of clarity, denote the wholesale price under the buyback contract and the revenue sharing contract as w_b and w_r, respectively. We can think of a buyback contract as requiring the retailer to pay $w_b - b$ per unit purchased and an additional b per unit sold. (This is equivalent to paying w_b per unit purchased and receiving b per unit unsold.) In a revenue sharing contract, the retailer pays $w_r + (1 - \phi)v$ per unit purchased and $(1 - \phi)(r - v)$ per unit sold. Then a revenue sharing contract with parameters w_r and ϕ is *equivalent* (in the sense that it generates the same profits no matter what the demand is) to a buyback contract if the parameters of the buyback contract satisfy

$$\begin{aligned} w_b - b &= w_r + (1 - \phi)v \\ b &= (1 - \phi)(r - v), \end{aligned}$$

that is,

$$w_b = w_r + (1 - \phi)r \tag{14.42}$$

$$b = (1 - \phi)(r - v). \tag{14.43}$$

However, in more complicated settings (for example, with more than one retailer), the two contracts are not equivalent.

□ **EXAMPLE 14.3**

Return to Example 14.1 and suppose that Jeffrey's Bakery offers a revenue sharing contract to Matilda's Market with $\phi = 0.4$. What w will coordinate the supply chain, and what are the resulting profits?

From (14.34),

$$w(0.4) = -0.25 + (0.75)\frac{0.4(1) + 0.2}{1 + 0.4} = 0.0714.$$

Since the contract coordinates, both players will choose $Q = Q^0 = 98.21$. From (14.35) and (14.39),

$$\pi_r(98.21, 0.0714, 0.4) = 0.6S(98.21) - 0.3214 \cdot 98.21 - 0.2 \cdot 100 = 3.0890$$
$$\pi_s(98.21, 0.0714, 0.4) = \Pi(Q^0) - \pi_r(98.21, 0.0714, 0.4) = 10.7854.$$

These are the same profits as under the buyback contract in Example 14.2. This is to be expected, since the parameters of the two contracts satisfy (14.42)–(14.43).

Other profit allocations are possible, of course. For example, if $\phi = 0.6803$ and $w = 0.2216$, then the retailer earns the entire profit, and if $\phi = 0.3917$ and $w = 0.0284$, the supplier does. □

14.8 THE QUANTITY FLEXIBILITY CONTRACT

We next introduce the *quantity flexibility contract* (Tsay 1999). The quantity flexibility contract is similar to the buyback contract in that the retailer pays a wholesale price per unit purchased and the supplier reimburses the retailer for unsold goods. The difference is that, in the buyback contract, the supplier *partially* reimburses the retailer for *every* unsold item, whereas in the quantity flexibility contract, she *fully* reimburses the retailer for a *portion* of his unsold items.

In particular, the quantity flexibility contract has two parameters, w and δ ($0 \le \delta \le 1$). The retailer pays the supplier w per unit ordered, and the supplier pays the retailer $(w + c_r - v)\min\{I, \delta Q\}$, where I is the on-hand inventory at the end of the period. Thus, the supplier agrees to protect the retailer against only a portion of his order: She will reimburse him for his losses on unsold merchandise (which equal $w + c_r - v$ per unit), but only up to a maximum of δQ units.

The quantity flexibility contract coordinates the supply chain from the retailer's end (his optimal order quantity is also the supply chain's optimal order quantity). However, unlike the contracts in Sections 14.6 and 14.7, the quantity flexibility contract only coordinates the supplier's decision for certain values of the parameters.

The transfer payment in the quantity flexibility contract is

$$T_q(Q, w, \delta) = wQ - (w + c_r - v)\left[\int_0^{(1-\delta)Q} \delta Q f(d)dd + \int_{(1-\delta)Q}^{Q} (Q - d)f(d)dd\right] \quad (14.44)$$

$$= wQ - (w + c_r - v)\int_{(1-\delta)Q}^{Q} F(d)dd. \quad (14.45)$$

(See Problem 14.10).

Let $w(\delta)$ be defined as

$$w(\delta) = \frac{(r - v + p_r)\bar{F}(Q^0)}{\bar{F}(Q^0) + (1-\delta)F((1-\delta)Q^0)} - c_r + v. \tag{14.46}$$

Theorem 14.7 *Under the quantity flexibility contract, for any* $0 \le \delta \le 1$, *if* $w(\delta)$ *is set according to* (14.46)*, then* $Q_r^* = Q^0$*, i.e., the supply chain is coordinated from the retailer's perspective.*

Proof. The retailer's profit function is

$$\begin{aligned}
\pi_r(Q, w, \delta) =& (r - v + p_r)S(Q) - (c_r - v)Q - p_r\mu \\
& - \left[wQ - (w + c_r - v) \int_{(1-\delta)Q}^{Q} F(d)dd \right] \\
=& (r - v + p_r)S(Q) - (w + c_r - v)Q - p_r\mu \\
& + (w + c_r - v) \int_{(1-\delta)Q}^{Q} F(d)dd.
\end{aligned} \tag{14.47}$$

In order for Q^0 to maximize $\pi_r(\cdot, w(\delta), \delta)$, it is necessary (but not sufficient) that $\partial\pi_r/\partial Q = 0$ when $Q = Q^0$. (It's not a sufficient condition because we also need to check that the second partial derivative is negative, i.e., that π_r is concave with respect to Q.)

$$\begin{aligned}
\frac{\partial \pi_r(Q, w(\delta), \delta)}{\partial Q} =& (r - v + p_r)S'(Q) - (w(\delta) + c_r - v) \\
& + (w(\delta) + c_r - v)\left[F(Q) + (1-\delta)F((1-\delta)Q)\right]
\end{aligned}$$

(using (C.48))

$$\begin{aligned}
=& (r - v + p_r)\bar{F}(Q) \\
& - (w(\delta) + c_r - v)[\bar{F}(Q) + (1-\delta)F((1-\delta)Q)].
\end{aligned} \tag{14.48}$$

When $Q = Q^0$, (14.48) equals 0 by (14.46).

It remains to show that π_r is concave when $w = w(\delta)$, i.e., that its second partial derivative is nonpositive.

$$\begin{aligned}
\frac{\partial^2 \pi_r(Q, w(\delta), \delta)}{\partial Q^2} =& -(r + p_r - w(\delta) - c_r)f(Q) \\
& - (w(\delta) + c_r - v)(1-\delta)^2 f((1-\delta)Q).
\end{aligned} \tag{14.49}$$

This is nonpositive if

$$v - c_r \le w(\delta) \le r + p_r - c_r,$$

which is true for all $0 \le \delta \le 1$ because

$$\begin{aligned}
w(0) &= (r - v + p_r)\bar{F}(Q^0) + v - c_r \ge v - c_r \\
w(1) &= r + p_r - c_r
\end{aligned}$$

and $w(\delta)$ is increasing in δ. Therefore, Q^0 maximizes $\pi_r(\cdot, w(\delta), \delta)$, provided that $w(\delta)$ is set according to (14.46). ∎

We also need to check whether Q^0 is optimal for the supplier; if it is not, the contract does not coordinate the supply chain. The supplier's profit function is

$$\pi_s(Q, w, \delta) = p_s S(Q) + (w - c_s)Q - p_s\mu - (w + c_r - v) \int_{(1-\delta)Q}^{Q} F(d)dd. \quad (14.50)$$

One can show that $\partial\pi_s/\partial Q$ *does* equal 0 when $Q = Q^0$. (See Problem 14.11.) However, for Q^0 to be a maximizer, the second partial derivative must be nonpositive. This derivative is

$$\frac{\partial^2 \pi_s(Q, w(\delta), \delta)}{\partial Q^2} = -w(\delta)\left[f(Q) - (1-\delta)^2 f((1-\delta)Q)\right] - p_s f(Q). \quad (14.51)$$

Unfortunately, this expression is not always nonpositive. For example, suppose $D \sim N(100, 25^2)$, $r = 10$, $c_r = c_s = 1$, and $p_r = p_s = v = 0$. If $\delta = 0.2$, then (14.51) equals -0.0028 (so Q^0 is a local max), but if $\delta = 0.1$, then (14.51) equals 0.0015 (so Q^0 is a local min).

All of this means that the quantity flexibility contract coordinates the supply chain from the retailer's point of view but not necessarily from the supplier's. In other words, when the retailer places an order of size Q^0, the supplier might wish to deliver an order of a different size Q'. The supplier can certainly not force the retailer to accept a larger order than he placed, so if $Q' > Q^0$ there's nothing the supplier can do about it. But if $Q' < Q^0$, the supplier wants to deliver an order smaller than the order placed by the retailer.

The attitude of the model toward this behavior is called the *compliance regime*. If the supplier is allowed to deliver less than the order size, the regime is called *voluntary compliance*. If the supplier is forced to deliver the entire order (because failing to do so would expose the supplier to a court action or to too much loss of goodwill, for example), it's called *forced compliance*. Since the supplier's optimal decision was also supply chain optimal in the coordinating contracts we've studied so far, the two regimes have been equivalent—the supplier wants to comply, even if she's not forced to do so. In the quantity flexibility contract, the supplier may not voluntarily comply.

Assuming that the supplier complies (either because she is forced to or because the parameters are such that her profit function is concave), the quantity flexibility contract, like the others, can allocate the profits in any way we like. (See Problem 14.12.)

☐ **EXAMPLE 14.4**

Return to Example 14.1 and suppose that Jeffrey's Bakery offers a quantity flexibility contract to Matilda's Market with $\delta = 0.15$. What w will coordinate the supply chain, and what are the resulting profits?

First note that

$$\bar{F}(Q^0) = 0.5357$$

$$F((1-\delta)Q^0) = F(83.4761) = 0.2044$$

$$\int_{(1-\delta)Q^0}^{Q^0} F(d)dd = \bar{n}(Q^0) - \bar{n}((1-\delta)Q^0) = 4.8193.$$

(The last equality follows from (C.13).) From (14.46),

$$w(0.15) = \frac{1.2 \cdot 0.5357}{0.5357 + 0.85 \cdot 0.2044} - 0.25 = 0.6562.$$

Since the contract coordinates, both players will choose $Q = Q^0 = 98.21$. From (14.47) and (14.50),

$$\begin{aligned}
\pi_r(98.21, 0.6562, 0.15) =& 1.2S(98.21) - 0.9062 \cdot 98.21 \\
& - 0.2 \cdot 100 + 0.9062 \cdot 4.8193 = 4.6844 \\
\pi_s(98.21, 0.6562, 0.15) =& 0.2S(98.21) + 0.1562 \cdot 98.21 \\
& - 0.2 \cdot 100 - 0.9062 \cdot 4.8193 = 9.1900
\end{aligned}$$

If $\phi = 0.0888$ and $w = 0.5450$, then the retailer earns the entire profit, and if $\phi = 0.1827$ and $w = 0.7123$, the supplier does. For all three values of ϕ discussed in this example, $\partial^2 \pi_s / \partial Q^2 < 0$, so Q^0 is a maximum of π_s for all three contracts, as desired. □

CASE STUDY 14.1 Designing a Shared-Savings Contract at McGriff Treading Company

Roughly half of all truck tires sold in the United States are retreaded tires ("retreads"), used tires in which the treads have been replaced with new ones. Retreads are less expensive than new tires, are more environmentally friendly, and have similar safety records. Trucking fleet operators use retreads as a way to reduce their tire-related costs, which constitute approximate 3% of their total operating costs (Yadav et al. 2003).

Both the retreading company and the fleet operator (trucking company) have ways to improve the useful life of a tire, as measured by the number of miles before the tire must be replaced. Retreaders can use higher-quality materials and enhancements to the production process, while fleet operators can provide financial incentives to drivers to exercise proper braking, turning, and tire inflation. Of course, in both cases, these measures come at a cost. Traditionally, retreaders sell retreads to fleet operators on a per-tire basis. This gives the retreader no explicit incentive to exert any effort to improve the durability of the tire. (Of course, there is an indirect incentive, since durability is one factor that fleet operators use when deciding whom to buy retreads from.) Another option that some retreaders have considered is to allow fleet operators to outsource all of their tire-related activities to the retreader, for a fixed annual fee. However, this arrangement has the opposite effect: Now, fleet operators have no incentive to expend any effort to reduce tire wear.

McGriff Treading Company wanted to find a contract structure that would give both parties an incentive to improve tire life, thus reducing costs for both parties. The company partnered with researchers at the University of Alabama to design such a contract structure. Their study is described by Yadav et al. (2003). At the time of the study, McGriff was one of the five largest independent truck tire retreaders in the United States, with roughly 200 employees and four retreading facilities, and a client list that included some of the world's largest fleet operators.

This is a different type of supply chain than those discussed earlier in this chapter: Rather than a supplier selling to a retailer facing random demand, we instead have a supplier selling an item to a company that uses it as an indirect material when providing a service that it sells to its customers. Thus, the contract has a different type of structure. Like the tire-outsourcing service described above, the tire retreader agrees to provide all of the fleet operator's tire-related service, for a fixed annual fee. (This is expressed in $/mile.) In addition, however, both parties split the savings that result from efforts to improve tire durability. The savings is measured by the difference between the retreader's total cost under the contract and its total cost if the fleet operator exerted no effort. The contract has two parameters, the per-mile price for tire-service outsourcing and the fraction of the savings earned by the fleet operator. Each player also has an additional decision variable representing the amount of effort expended to reduce tire wear.

Let s and r represent the the retreading company (the supplier) and the fleet operator (not really a retailer, but we will use the symbol r anyway, for consistency). Let c_s be the retreading company's cost per tire and let w be the per-unit cost that the retreader charges the fleet operator; we'll assume $c_s < w$. The players each decide the effort they will exert to reduce tire wear; this effort is a decision variable and is denoted e_s and e_r for the retreader and fleet operator, respectively. Effort levels e_s and e_r cost the players $\kappa_s(e_s)$ and $\kappa_r(e_r)$, respectively, and both functions are increasing. These effort levels result in an annual usage of $u(e_s, e_r)$ tires per vehicle mile, which is decreasing and concave in both e_s and e_r. Assume that the fleet operator earns an annual revenue of R, which is independent of the decisions made in the model but is used to formulate the profit function. Finally, let F be the fixed fee, expressed in $/mile, that the retreader charges the fleet operator for tire-service outsourcing, and let ω be the fraction of the savings that the fleet operator earns; these are the parameters of the contract.

Under the traditional selling arrangement, in which fleet operators simply buy tires at a per-unit cost, the two parties have profit functions

$$\pi_s^0(e_s, e_r) = (w - c_s)u(e_s, e_r) - \kappa_s(e_s) \tag{14.52}$$

$$\pi_r^0(e_s, e_r) = R - wu(e_s, e_r) - \kappa_r(e_r). \tag{14.53}$$

(Note that these are deterministic profit functions since there are no random variables in this model.) Since $c_s < w$, $u(e_s, e_r)$ is decreasing in e_s, and $\kappa_s(e_s)$ is increasing, it is optimal for the retreader to exert no effort: $e_s^0 = 0$. (The fleet operator's optimal effort is found by setting $\partial \pi_r^0 / \partial e_r = 0$, but we will not need this quantity.)

Now consider the tire-service outsourcing contract, but without the shared-savings component, i.e., $\omega = 0$. The profit functions are given by

$$\pi_s^n(e_s, e_r, F) = F - c_s u(e_s, e_r) - \kappa_s(e_s) \tag{14.54}$$

$$\pi_r^n(e_s, e_r, F) = R - F - \kappa_r(e_r). \tag{14.55}$$

(The superscript n denotes "no sharing.") In this case, the fleet operator has no incentive to exert effort, since R and F are constants and $\kappa_r(e_r)$ is increasing; thus $e_r^n = 0$. We can find the retreader's optimal effort by setting $\frac{\partial \pi_s(e_s, 0)}{\partial e_s} = 0$, but we will not need this value either.

Finally, consider the tire-service outsourcing contract with shared savings. The total savings is defined as

$$\Delta(e_s, e_r) = c_s \left[u(e_s^n, 0) - u(e_s, e_r) \right],$$

that is, the difference between the retreader's cost under the shared-savings contract and the no-shared-savings contract. Since a fraction ω of the savings goes to the fleet operator and $1-\omega$ goes to the retreader, the profit functions are then given by

$$\pi_s^s(e_s, e_r, F, \omega) = F - c_s u(e_s, e_r) - \kappa_s(e_s) + (1-\omega)\Delta(e_s, e_r) \tag{14.56}$$

$$\pi_r^s(e_s, e_r, F, \omega) = R - F - \kappa_r(e_r) + \omega\Delta(e_s, e_r). \tag{14.57}$$

(The superscript s is for "sharing.") For a given value of F, we can find the value $\omega(F)$ that satisfies the first-order conditions; then we can use the first-order conditions again to find the F that the retreader should select.

In practice, McGriff does not have all of the data (R, $\kappa(\cdot)$, etc.) required to calibrate this model precisely. Therefore, the firm opted to present potential clients with a menu of contract options consisting of a few values of F and the corresponding ω values.

PROBLEMS

14.1 (Existence of Q_s^* under Wholesale Price Contract) Prove that, under the wholesale price contract, $\pi_s(Q, w)$ is:

- strictly decreasing in Q if $w < c_s - p_s$
- first increasing and then decreasing in Q if $c_s - p_s < w < c_s$
- strictly increasing in Q if $w > c_s$.

14.2 (Wholesale Price Contract for *Breach of Contract*) A new novel, the legal thriller *Breach of Contract*, will begin to be sold at your local bookstore next month. The bookstore must decide how many copies of the book to order, and it cannot reorder again after the initial order. The book will be sold for a certain duration (say, 6 months), after which all copies will be removed from the shelves and sold to a paper-recycling company as scrap.

Breach of Contract will be sold to consumers for \$18.99 per copy. The publisher charges the bookstore a wholesale price of \$11.00 per copy. For each copy purchased by the bookstore, the publisher incurs raw-material costs of \$3.75, and the bookstore incurs shipping and handling costs of \$1.20. (This is not paid to the publisher.) The total demand for the book during the selling season is expected to be normally distributed with a mean of 1200 and a standard deviation of 340. Unmet demands are lost, incurring loss-of-goodwill costs estimated at \$9.00 for the bookstore and \$4.00 for the publisher. Unsold books are sold to the recycling company for \$0.65 each.

a) What is the bookstore's optimal order quantity?

b) What is the order quantity that maximizes the total expected profit for the supply chain, and what is the optimal total expected profit?

c) What is each company's expected profit, and what is the total expected profit for the supply chain? What is the efficiency of the contract?

d) Suppose the publisher can choose any wholesale price it wishes. What wholesale price will it choose? What order quantity will the bookstore choose as a result? What will be the resulting profits? What will be the efficiency of the contract?

14.3 (Buyback Contract for *Breach of Contract*) Consider the supply chain discussed in Problem 14.2. Suppose the publisher offers the bookstore a buyback contract with a buyback credit of \$8. (Buyback contracts are common in the publishing industry. However, since it is expensive to ship books, it is common practice for the publisher to require the bookstore to return only the cover of the book, and to destroy the rest of the book (Chopra and Meindl 2001).)

a) What w will coordinate the supply chain, and what will be the resulting profits? What is the supplier's percentage of the profit?

b) What value of b will give the retailer all of the profit? What value will give the supplier all of the profit?

14.4 **(Revenue Sharing Contract for *Breach of Contract*)** Consider the supply chain discussed in Problem 14.2. Suppose the publisher offers the bookstore a revenue sharing contract in which the bookstore keeps 60% of its revenue and gives 40% to the publisher.

a) What w will coordinate the supply chain, and what will be the resulting profits? What is the supplier's percentage of the profit?

b) What value of ϕ will give the retailer all of the profit? What value will give the supplier all of the profit?

14.5 **(Quantity Flexibility Contract for *Breach of Contract*)** Consider the supply chain discussed in Problem 14.2. Suppose the publisher offers the bookstore a quantity flexibility contract with $\delta = 0.4$.

a) What w will coordinate the supply chain, and what will be the resulting profits? What is the supplier's percentage of the profit? Does the supply-chain-optimal order quantity also maximize the publisher's profit function?

b) What value of δ will give the retailer all of the profit? What value will give the supplier all of the profit?

14.6 **(Retailer Under-Orders)** Prove Theorem 14.2.

14.7 **(Redundancy of (14.19))** Prove that if b and w satisfy (14.18) and (14.22), then they also satisfy (14.19).

14.8 **(Theater Ticket Returns Policies)** A school is planning a class trip for its first-grade students to see a play at a local children's theater. There are 85 students in the first grade. The school will buy tickets in advance for \$10 each. On the day of the play, if some children are sick and absent from school, the theater will not allow the unused tickets to be returned or exchanged. Therefore, the school is planning to buy $Q < 85$ tickets. However, if more than Q students show up to school on the day of the play, some of the children will have to stay at school, incurring a child-care cost (paid by the school) of \$13 per student. Assume that a given student will be absent from school with probability 0.05, and that absences are statistically independent across students.

a) What is the optimal number of tickets for the school to purchase? What will be the theater's revenue (ticket sales)?

Hint: Use the normal approximation to the binomial distribution. You may assume that fractional ticket sales are possible.

b) Suppose the theater implements a policy under which they will refund the school \$6 for each unused ticket. Now what is the optimal number of tickets for the school to purchase? What will be the theater's expected net revenue (ticket sales minus refunds)?

14.9 (Buyback Parameters Are Valid) Prove that, for any revenue sharing contract (w_r, ϕ), the equivalent buyback parameters defined in (14.42)–(14.43) satisfy (14.22).

14.10 (Quantity Flexibility Transfer Payment) Prove (14.45), taking (14.44) as given.

14.11 (First-Order Condition for Quantity Flexibility) Prove that

$$\left.\frac{\partial \pi_s(Q, w(\delta), \delta)}{\partial Q}\right|_{Q=Q^0} = 0$$

in the quantity flexibility contract. That is, Q^0 is a stationary point for the supplier's profit function.

14.12 (Profit Allocation Under Quantity Flexibility Contract) Prove that, under the quantity flexibility contract, for any $0 \le \alpha \le 1$, there exists a δ such that the supplier receives exactly α proportion of the total profit; that is, all possible allocations of the profit are possible.

14.13 (Alternate Proof of Theorem 14.6) Prove Theorem 14.6 using a method similar to Proof #1 of Theorem 14.4; that is, use the first-order conditions directly to show that $Q_r^* = Q_s^* = Q^0$.

14.14 (A Simpler Contract?) A student once made a comment along the lines of, "Why bother with this contracting stuff? If the supply chain profit is not maximized when the retailer orders Q_r^* instead of Q^0, why don't the parties just agree that the retailer will order Q^0 and then they'll split the extra profits somehow?"

In essence, this student has proposed an alternate, potentially simpler, contracting mechanism. Suppose the parties decide to split the profits by bringing the retailer's profit up to the profit he'd earn if he'd ordered Q_r^* instead of Q^0. That is, the supplier agrees to pay the retailer $\pi_r(Q_r^*) - \pi_r(Q^0)$ per period if the retailer orders Q^0. In addition, let's assume there's a wholesale price of w per unit, which is fixed (not a parameter of the contract), and that $w > c_s$.

In other words, the transfer payment T is given by

$$T = \begin{cases} wQ - (\pi_r(Q_r^*) - \pi_r(Q^0)), & \text{if } Q = Q^0 \\ wQ, & \text{if } Q \neq Q^0, \end{cases}$$

where π_r is as defined in (14.9).

a) Write the retailer's expected profit function under this simple contract (call it $\pi_r^s(Q)$). You may express $\pi_r^s(Q)$ in terms of $\pi_r(Q)$.

b) Prove that, under this contract, the retailer is indifferent between ordering Q^0 and Q_r^*—both maximize his expected profit.

c) Write the supplier's expected profit function under the new contract (call it $\pi_s^s(Q)$). You may express $\pi_s^s(Q)$ in terms of $\pi_r(Q)$ and $\pi_s(Q)$, where $\pi_s(Q)$ is as defined in (14.10) .

d) Prove that the supplier prefers the retailer to order Q^0 (instead of Q_r^*) if and only if

$$\pi_s(Q^0) - \pi_s(Q_r^*) > \pi_r(Q_r^*) - \pi_r(Q^0).$$

(If the supplier prefers Q_r^*, then the contract fails to coordinate the supply chain, since the supplier wouldn't even propose the contract to begin with.)

e) Prove that the condition in part (d) always holds.

14.15 (Second Ordering Opportunity) Consider a supply chain with a single supplier and a single retailer. The retailer has two opportunities to order items from the supplier: once before he knows the actual demand and once after. All demands must be satisfied; therefore, the size of the second order is equal to any shortfall from the first order. However, any demands that are not met after the first order incur a loss-of-goodwill cost p per item since customers will have to wait until the second order is placed before receiving their products.

Demands are random with pdf f, cdf F, and mean μ. Let Q be the size of the first order. The selling price is r per item regardless of when the demand is satisfied. The supplier charges the retailer a wholesale price of w per item for both the first and second orders. Unsold merchandise at the end of the period may be salvaged for a salvage value of v per unit.

The manufacturer produces to order; that is, she produces exactly the number of units requested by the retailer in each order and does not hold inventory between orders. She incurs a production cost of c_1 for items produced for the first order and c_2 for items produced for the second order. Since the second order typically requires smaller production runs, you can assume $c_1 < c_2$. You can also assume that $v < c_1$ and $c_2 < w$.

The sequence of events in the time period is as follows: The retailer places his first order. The order is delivered immediately. Demand is realized, and all demands that can be met from stock are satisfied; the remaining demands are put on hold until the second order. If any demands are on hold, the second order is placed (for exactly the number of units on hold). The second order arrives immediately, and the on-hold demands are satisfied. If the demand was smaller than the first order, any unsold items are salvaged.

a) Write expressions for the retailer's, supplier's, and supply chain's expected profit as a function of Q, denoted $\pi_r(Q)$, $\pi_s(Q)$, and $\Pi(Q)$, respectively.

Hint: To check that you have the correct formulas before you use them in the subsequent parts, we'll tell you the following: Assuming that

$$\begin{aligned} r &= 100 & w &= 50 \\ p &= 125 & c_1 &= 25 \\ v &= 20 & c_2 &= 35 \\ Q &= 300 \end{aligned}$$

and demand is distributed $N(200, 50^2)$, then

$$\begin{aligned} \pi_r(Q) &= 6934.2 \\ \pi_s(Q) &= 7506.4 \\ \Pi(Q) &= 14,440.6. \end{aligned}$$

b) Prove that the retailer's optimal order quantity is strictly smaller than the supply chain's optimal order quantity. (Therefore, the supply chain is not coordinated.)

c) Consider a buyback contract in which the retailer pays the supplier a wholesale price of w (replacing the wholesale price w used in the original model) and the supplier pays the retailer a subsidy of b for every unit of unsold merchandise at the end of the period. Prove that if the wholesale price is given as a function of b

by

$$w(b) = b + v + p\frac{c_1 - v}{p + c_2 - c_1},$$

then the supply chain is coordinated. Make sure you verify all relevant necessary and sufficient conditions.

d) In a few sentences, explain why the original supply chain was not coordinated, and why the buyback contract coordinates the supply chain.

14.16 **(Completing the Proof of Theorem 14.3)** Fill in the missing step in the proof of Theorem 14.3 by showing that

$$\lim_{Q\to\infty} 1 + \frac{p_s}{r - v + p_r} - \frac{Qf(Q)}{\bar{F}(Q)} = -\infty.$$

You may assume that the demand is normally distributed and use the fact that for a $N(\mu, \sigma^2)$ distribution, $f'(x) = -f(x)\frac{x-\mu}{\sigma}$.

CHAPTER 15

AUCTIONS

15.1 INTRODUCTION

Auctions have been around for centuries and the mathematical analysis of auctions dates back decades. But they have enjoyed growing popularity in recent years because the Internet has made efficient implementation of auctions, even complex ones, possible. Consumer auctions like eBay have become household names, but business-to-business (B2B) auctions have grown even more quickly. B2B auctions are mainly procurement auctions in which there is a single buyer and multiple sellers (the reverse of most consumer auctions; in fact, such auctions are called *reverse auctions*). For example, auto manufacturers have set up auctions in which thousands of potential suppliers bid for contracts; the auto company chooses the suppliers with the lowest prices. Clearly, such an undertaking would be much more cumbersome without the Internet. We will consider auctions with a single seller and multiple potential buyers.

There are many types of auctions, each with various properties in terms of consumer behavior, efficiency, and so on. Here are just a few types of auctions that have been introduced in the literature and in practice:

- *English*: Perhaps the most familiar type, with each bidder publicly announcing his bid and the price rising until only a single bidder remains. The highest bidder wins and pays his bid.

Fundamentals of Supply Chain Theory, Second Edition. Lawrence V. Snyder and Zuo-Jun Max Shen.

Companion website: www.wiley.com/go/Snyder/SupplyChainTheory

- *Sealed-bid first price*: Bids are made privately and simultaneously by all bidders. The bidder with the highest bid wins and pays his bid.

- *Sealed-bid second price*: Sometimes known as a *Vickrey* auction, this auction type is like the sealed-bid first price auction except that the winner pays the *second-highest* bid, not his own bid (though the bidder with the highest bid is still the winner). Second price auctions encourage higher bidding since the winning bidder pays a lower price than his own bid.

- *Dutch*: In the Dutch auction, the price starts high and the auctioneer announces successively lower prices. As soon as one bidder accepts the current price, that bidder wins and pays the price, and the auction ends.

In addition to deciding the auction type, the auction designer (who may be the buyer, seller, or a third party like eBay) must decide aspects of the auction structure like how bidders may bid on multiple units (e.g., as a package or individually), what information is available to the bidders (e.g., the bids that have been placed by other bidders), and so on.

Auctions can be seen as mechanisms for supply chain coordination since they give buyers and sellers an opportunity to negotiate a mutually beneficial agreement. Game-theoretic issues appear in auction analysis, too; for example, players may have an incentive to misrepresent their objectives through misleading bids.

In fact, there are several important properties that are desirable for auctions to have. These desirable properties include:

- *Strategy-proof*: In a strategy-proof auction, truthful bidding is never worse than untruthful bidding, for each buyer and seller. A related, but weaker, concept is *incentive-compatibility*, which means that truthful bidding is a (Bayesian) Nash equilibrium.

- Ex post *individually rational*: An auction is *ex post* individually rational if each buyer and each seller do at least as well if they participate in the auction (under any auction outcome) than if they don't participate.

- Ex post *budget-balanced*: An auction is *ex post* budget-balanced if the auctioneer's payoff is nonnegative for all possible outcomes; therefore, the auctioneer can hold the auction without an outside subsidy.

- *Optimal or efficient*: An *optimal* auction implements an allocation that maximizes expected revenue (the sum of the expected payments of the buyers), while an *efficient* auction maximizes social welfare (i.e., achieves the highest possible set of awarded valuations).

Any mechanism must be individually rational and budget balanced to make an auction practical. Moreover, strategy-proofness is desired, since each agent may not know enough information about the other agents to determine his or her optimal strategy. Unfortunately, it is not possible to design an auction mechanism that is efficient, individually rational, incentive compatible, and budget balanced—at least one of these must be sacrificed (Myerson and Satterthwaite 1983).

In Section 15.2, we will analyze a simple English auction and show that the auction itself can be thought of in terms of a linear program and its dual. Then, in Section 15.3

we will discuss a more complicated auction with multiple products and investigate the allocation problem faced by the auctioneer. Finally, in Section 15.4, we analyze the famous Vickrey–Clarke–Groves (VCG) auction.

15.2 THE ENGLISH AUCTION

In an English auction, there is a set of bidders (also called *agents*), each bidding on a single item. The price begins low and gradually increases. At each price announced by the auctioneer, all bidders announce whether they are still willing to bid on the item at the current price (for example, by raising their hands), and the auction ends when only a single bidder remains. In this section, we analyze such an auction. Our analysis is adapted from Kalagnanam and Parkes (2004).

Let N be the set of agents. Agent $i \in N$ has a valuation v_i that she has assigned to the item: v_i is the maximum she'd be willing to pay for it. Paying v_i is like breaking even, so she'd prefer to pay less. The auctioneer knows v_i for each bidder. His goal is to award the item to the bidder with the highest valuation. (This also maximizes the auctioneer's revenue, assuming he is the seller. However, under this auction, the auctioneer will not necessarily receive v_i for the winning bidder.)

In other words, the auctioneer needs to solve the following problem:

$$
\begin{aligned}
\text{(IP)}\quad \text{maximize} \quad & \sum_{i \in N} v_i x_i && (15.1)\\
\text{subject to} \quad & \sum_{i \in N} x_i \le 1 && (15.2)\\
& x_i \in \{0,1\} \qquad \forall i \in N && (15.3)
\end{aligned}
$$

where $x_i = 1$ if agent i is awarded the item. The constraint says that at most one agent may be assigned the item. The English auction is one way of solving this problem. Another way is simply to ask each bidder for his or her valuation and to award the item to the bidder with the highest valuation, but bidders may prefer the English auction since it allows them to win the item without paying their maximum valuation.

The following problem is also equivalent to (IP):

$$
\begin{aligned}
\text{(LP)}\quad \text{maximize} \quad & \sum_{i \in N} v_i x_i && (15.4)\\
\text{subject to} \quad & \sum_{i \in N} x_i + y = 1 && (15.5)\\
& x_i \le 1 \qquad \forall i \in N && (15.6)\\
& x_i, y \ge 0 \qquad \forall i \in N && (15.7)
\end{aligned}
$$

In this formulation, a new variable y is added that represents the auctioneer not assigning the item to any bidder. Then the constraint can be written as an equality instead of an inequality. Furthermore, in this formulation the integrality restriction has been dropped. We can do this because although problem (IP) is an integer program, it always has an optimal solution in which the x_i are integer. Therefore, it is equivalent to its LP relaxation.

Now consider the LP dual of (IP), with p the dual variable for constraint (15.5) and π_i the dual variable for constraint (15.6):

$$\text{(D)} \quad \text{minimize} \quad p + \sum_{i \in N} \pi_i \tag{15.8}$$

$$\text{subject to} \quad \pi_i \geq v_i - p \qquad \forall i \in N \tag{15.9}$$

$$p \geq 0 \tag{15.10}$$

$$\pi_i \geq 0 \qquad \forall i \in N \tag{15.11}$$

Note that p is nonnegative (constraint (15.10)) since the coefficient of y in (15.4) is 0, while the nonnegativity of π_i (constraints (15.11)) follows from the inequality constraints (15.6).

Suppose p is fixed. The optimal values of π_i are given by $\pi_i = \max\{0, v_i - p\}$. A primal solution x and a dual solution (p, π) are optimal for their respective problems if the complementary slackness conditions hold:

$$p > 0 \implies \sum_i x_i + y = 1 \tag{CS1}$$

$$\pi_i > 0 \implies x_i = 1 \quad \forall i \in N \tag{CS2}$$

$$x_i > 0 \implies \pi_i = v_i - p \quad \forall i \in N \tag{CS3}$$

$$y > 0 \implies p = 0 \tag{CS4}$$

The dual variables p and π_i have a natural interpretation in the context of the auction: p is the selling price of the item and π_i is the payoff (valuation minus price) to agent i under price p. The complementary slackness conditions are then interpreted as follows:

- If the price is positive, then by CS1 either someone gets it or no one gets it. By CS4, if no one gets it, then the price must be 0. So taken together, CS1 and CS4 mean if the price is positive, someone gets the item.

- By CS3, if agent i wins the auction, then the price must equal $v_i - \pi_i$; since $\pi_i \geq 0$ by (15.11), this means the winning agent pays no more than his valuation.

- By CS2, any agent not receiving the item ($x_i = 0$) must have $\pi_i = 0$; this means $v_i \leq p$ by (15.9), i.e., the price is greater than the losing agent's valuation.

Notice that these are exactly the conditions under which the auction ends.

The simplex method is called a *primal algorithm* for solving LPs because it maintains a primal solution at all times and tries to improve it until it finds the optimal solution. Other methods, called *primal–dual algorithms*, maintain both a primal and a dual solution at all times and attempt to move toward optimal solutions by fixing violations in the complementary slackness conditions.

This is exactly the process taken by the English auction! In a sense, the auction is nothing more than a big LP solver in which the actions of the players correspond to steps in the algorithm. In particular, interpret the dual variable p as the current price and the dual variables π_i as the corresponding payoffs for each agent. Interpret the primal variable x_i as indicating whether agent i is the current "provisional" winner of the auction, chosen arbitrarily from among the bidders that are still interested in the item at the current price p. Interpret y as indicating that no bidders are still interested in the item at the current price. Throughout the auction, the primal and dual solutions are both feasible for their respective problems. When the complementary slackness conditions hold, the auction ends, the optimal solutions having been found. Note that in every round, CS1, CS3, and CS4 hold in the auction: If the price is positive, some agent must be the current provisional winner (CS1 and CS4), and the price is less than the value of the provisional winner (CS3). CS2 might not hold since there may be nonwinners who still have a positive payoff, that is, whose valuations are less than the current price. The primal-dual approach works by increasing p until CS2 holds.

15.3 COMBINATORIAL AUCTIONS

English auctions involve only a single item for sale. In this section, we will discuss auctions in which there are a number of heterogeneous items for sale. In such an auction, bidders may want to bid on combinations of items instead of individual items. For example, suppose you attend an auction of some antique furniture. You might be interested in buying the bed and matching dresser if you could buy them together, but might not be interested in buying only one of them. Or, you might be interested in buying one bed or another, but not both. Valuation, then, is assigned to subsets of items (called *bundles*) rather than to individual items. This makes the auction itself, as well as the auctioneer's allocation problem, considerably more complex. Such auctions are called *combinatorial auctions*. Our analysis of combinatorial auctions is adapted from de Vries and Vohra (2003). For further discussion of combinatorial auctions in practice, see Harstad and Pekeč (2008).

A famous example of a combinatorial auction is the occasional auction of telecommunications spectrum rights held by the US Federal Communications Commission (FCC). A cell phone carrier, for example, might want to buy one license in each market it's interested in. Bidding for individual licenses misses the point since the value of one license depends on which other licenses the company holds. In the FCC's first auction, in 1994, they allowed bids on individual licenses only (though steps were taken to help bidders obtain bundles they were interested in), thinking it would be too cumbersome to allow bids on bundles. However, in 2003, the FCC held its first auction in which bidders could bid on combinations of licenses.

Another example of a practitioner of combinatorial auctions is JUNAEB, the agency responsible for providing free meals to low-income schoolchildren in Chile. Since 1980, the agency has used auctions to select private companies to provide these meals throughout the country. Companies place bids that indicate the geographic region they will serve, the services they will provide, and the price they will charge. Prior to 1997, bids were chosen more or less independently, without considering the interdependencies among the bids. But in 1997, JUNAEB began using combinatorial auctions to allocate bids. Each company can submit multiple bids, for example, covering different combinations of geographical regions. The new auction mechanism saves JUNAEB an estimated US$40 million per year and has also improved the quality of the food, the geographic scope of the assistance program, and the transparency of the entire bidding process (Epstein et al. 2004).

15.3.1 The Combinatorial Auction Problem

Let M be the set of objects being auctioned off, and let N be the set of bidders. For any bundle $S \subseteq M$, let b_{iS} be the bid that agent i has announced he is willing to pay for S.

Note that b is different from v since it represents announced bids, not valuations; an agent's bid for a bundle might be less than his valuation. In an auction of any reasonable size, it would be impossible for an agent to specify a bid for all $2^{|M|}$ possible bundles, so you can think of b as a function that takes a bundle suggested by the auctioneer and returns a bid for that bundle. Without loss of generality we can assume $b_{iS} \geq 0$ for all i, S. Let y_{iS} be 1 if agent i is assigned bundle $S \subseteq M$ and 0 otherwise.

The auctioneer's problem is to allocate the bundles to agents in order to maximize her revenue. This problem is known as the *combinatorial auction problem* (CAP) and is formulated as follows:

$$\text{(CAP)} \quad \text{maximize} \quad \sum_{i \in N} \sum_{S \subseteq M} b_{iS} y_{iS} \tag{15.12}$$

$$\text{subject to} \quad \sum_{S \ni j} \sum_{i \in N} y_{iS} \leq 1 \qquad \forall j \in M \tag{15.13}$$

$$\sum_{S \subseteq M} y_{iS} \leq 1 \qquad \forall i \in N \tag{15.14}$$

$$y_{iS} \in \{0, 1\} \qquad \forall i \in N, \forall S \subseteq M \tag{15.15}$$

The objective function maximizes the revenue to the seller. (If the bids are equal to the actual valuations, then this formulation also maximizes the "efficiency" of the auction—assigning bundles to the agents that value them the highest.) Constraints (15.13) ensure that no two bundles assigned to agents contain the same item. (The summation over $S \ni j$ is a summation over all $S \subseteq M$ that contain j.) Constraints (15.14) prevent an agent from receiving more than one bundle. This is necessary to ensure that the auctioneer does not decide to assign bundles S and T to bidder i, instead of $S \cup T$, if $b_{iS} + b_{iT} > b_{i,S \cup T}$. However, we will restrict ourselves to cases in which a bid for a bundle is no smaller than the sum of bids of subsets of the bundle; that is, $b_{i,S \cup T} \geq b_{iS} + b_{iT}$ for all $S, T \subseteq M$. Bid functions for which this property holds are called *superadditive*. If the bid functions are superadditive, then constraints (15.14) are no longer needed since it is always to the seller's advantage to award an agent a single bundle rather than two separate ones.

We now reformulate (CAP) as an instance of the *set packing problem* (SPP), in which there is a set M of elements and a collection V of subsets of M with nonnegative weights; the objective is to choose the largest-weight collection of subsets such that every element is contained in at most one subset. (The SPP is like the inverse of the set covering problem (Section 8.4.1), in which we want to choose subsets such that every element is contained in at least one subset. If every element must be contained in exactly one subset, we have the set-partitioning problem.)

Let V be the collection of all subsets of M, and let k be an individual bundle (subset). Define

$$b_k = \max_{i \in N} \{b_{ik}\},$$

that is, b_k is the maximum bid any agent would be willing to pay for bundle k. Let $a_{ik} = 1$ if bundle k contains item i, 0 otherwise. Finally, let $x_k = 1$ if bundle k is selected, 0 otherwise. The problem now reduces to one of partitioning the items into bundles; the actual assignments can be done after the fact based on which agent maximized b_{ik} for each selected bundle k. The CAP can be reformulated as:

$$\text{(SPP)} \quad \text{maximize} \quad \sum_{k \in V} b_k x_k \tag{15.16}$$

$$\text{subject to} \quad \sum_{k \in V} a_{jk} x_k \leq 1 \qquad \forall j \in M \tag{15.17}$$

$$x_k \in \{0, 1\} \qquad \forall k \in V \tag{15.18}$$

Constraints (15.17), like constraints (15.13), prohibit the bundles from overlapping. We will focus on this formulation of the CAP in what follows.

15.3.2 Solving the Set-Packing Problem

Unlike the auctioneer's problem in the English auction, the CAP does not naturally have all integer solutions; that is, it is not equivalent to its LP relaxation. Therefore, we can't use LP duality to solve the problem. However, we will use Lagrangian relaxation (which provides a different type of duality) to solve it and show that, like the LP dual, the Lagrangian formulation has a natural interpretation in the auction context.

Let's relax constraints (15.17) in the SPP using Lagrange multipliers λ. The resulting subproblem is

$$\begin{aligned} (\text{SPP-LR}_\lambda) \quad z_{\text{LR}}(\lambda) = \text{maximize} \quad & \sum_{k \in V} b_k x_k + \sum_{j \in M} \lambda_j \left(1 - \sum_{k \in V} a_{jk} x_k\right) \\ = & \sum_{k \in V} \left(b_k - \sum_{j \in M} \lambda_j a_{jk}\right) x_k + \sum_{j \in M} \lambda_j \qquad (15.19) \\ \text{subject to} \quad & x_k \in \{0, 1\} \qquad \forall k \in V \qquad (15.20) \end{aligned}$$

Solving (SPP-LR_λ) is easy: For each $k \in V$ we set $x_k = 1$ if

$$b_k - \sum_{j \in M} \lambda_j a_{jk} > 0$$

and 0 otherwise. Since (SPP) is a maximization problem, for a given λ, $z_{\text{LR}}(\lambda)$ provides an *upper bound* on that of (SPP), and our goal is to find better (i.e., smaller) upper bounds by solving

$$\text{(SPP-LR)} \quad \min_{\lambda \geq 0} \quad z_{\text{LR}}(\lambda).$$

Note that λ is restricted to be nonnegative; see Section D.1.5. Problem (SPP-LR) is sometimes known as the *Lagrangian dual*, because in many ways it behaves like an LP dual. (SPP-LR) can be solved approximately using subgradient optimization, as we did when using Lagrangian relaxation in Chapter 8. A solution to (SPP-LR_λ) can be converted into a feasible solution for (SPP) using some heuristic; this feasible solution then provides a lower bound.

Our interest is not so much in this solution method as in its auction interpretation. The Lagrange multiplier λ_j represents the price on an individual item set by the auctioneer. The agents have already announced their bids for each bundle $k \in V$, the maximum of which is b_k. If $b_k > \sum_{j \in M} a_{jk}\lambda_j$, then, according to the solution to (SPP-LR$_\lambda$), x_k is set to 1—the bundle is temporarily included in the group of bundles to be sold. Presumably, the bundles in that group may overlap (constraints (15.17) may be violated), in which case the auctioneer must adjust the prices λ using subgradient optimization. Prices for items that are included in too many bundles would increase, while prices for items that are in no bundles would decrease.

As in the English auction, feasible solutions to the dual problem represent prices, while feasible solutions to the primal problem represent tentative assignments of items to agents. A primal-dual pair is optimal if something akin to the complementary slackness conditions hold:

- If $\lambda_j > 0$, then $\sum_{k \in V} a_{jk}x_k = 1$, i.e., item j is contained in exactly one allocated bundle. Another way of saying this is that if j is not included in any bundle, then it is a worthless item, so $\lambda_j = 0$.

- If $x_k > 0$, then $\sum_{j \in M} \lambda_j a_{jk} \leq b_k$, i.e., the bundle is worth at least the asking price to some bidder.

15.3.3 Truthful Bidding

In the combinatorial auction described above, the seller allocates bundles to maximize her revenue based on the bids. But there is no guarantee that the bids accurately reflect the bidders' valuations of the items, and bidders may have an incentive to lie. For example, suppose there are three bidders (1, 2, and 3) and two items (A and B). The bidders' valuations of the three possible bundles are given in Table 15.1.

Table 15.1 Valuations that induce nontruthful bidding.

	Bundle		
Bidder	A	B	A,B
1	0	0	100
2	75	75	0
3	40	40	0

The bidders do not need to place bids equal to their valuations. If they *did* bid truthfully, the auctioneer should award A to 2 and B to 3 (or vice-versa), for a revenue of 115. However, if bidder 2 assumes that bidder 3 will continue to bid truthfully, he has an incentive to reduce his bid on A and B, say, to 65. Bidders 2 and 3 still win the auction, but bidder 2 pays less. Bidder 3 can reason the same way—but if they both reduce their bids, bidder 1 might win the auction.

If bidders 2 and 3 could collude on their bids, they could ensure that they win the auction, as long as the sum of their bids (for each item) is greater than 100, though this leaves a profit of 15 that they need to decide how to share.

Obviously, it is to the auctioneer's advantage if the agents bid truthfully. She can't force them to do so, but she can design the auction mechanism so that the bidders' optimal strategy is to bid truthfully. This is very similar to the supply chain contracts discussed in Chapter 14: By structuring the payoffs carefully, the mechanism designer motivates the players to behave in a particular way, even when they are acting selfishly. In the case of auctions, however, the auctioneer is trying to manipulate the game so that the agents maximize her own revenue, while in supply chain contracting, the contracts are designed to maximize the common revenue.

15.4 THE VICKREY–CLARKE–GROVES AUCTION

15.4.1 Introduction

In this section, we discuss an auction mechanism in which it is to the agents' benefit to bid truthfully. The auction is called the *Vickrey–Clarke–Groves* (VCG) auction, after the researchers who proposed and studied it. The VCG auction is a single-round sealed-bid auction. (If there is only a single item, it is equivalent to a sealed-bid second-price auction.) It works as follows:

1. Agent i reports his valuation v_{iS} for all $S \subseteq M$. There is nothing to prevent the agents from misreporting their valuations, but it turns out to be optimal for them to be truthful.

2. The auctioneer solves the following problem:

$$
\begin{aligned}
V = \text{maximize} \quad & \sum_{i \in N} \sum_{S \subseteq M} v_{iS} y_{iS} \\
\text{subject to} \quad & \sum_{S \ni j} \sum_{i \in N} y_{iS} \le 1 && \forall j \in M \\
& \sum_{S \subseteq M} y_{iS} \le 1 && \forall i \in N \\
& y_{iS} \in \{0,1\} && \forall i \in N, \forall S \subseteq M
\end{aligned}
$$

   Note that this is just (CAP) with b replaced by v. Let y^* be the optimal solution.

3. The auctioneer solves the following problem for each agent $k \in N$:

$$
\begin{aligned}
V^{-k} = \text{maximize} \quad & \sum_{i \in N \setminus k} \sum_{S \subseteq M} v_{iS} y_{iS} \\
\text{subject to} \quad & \sum_{S \ni j} \sum_{i \in N \setminus k} y_{iS} \le 1 && \forall j \in M \\
& \sum_{S \subseteq M} y_{iS} \le 1 && \forall i \in N \setminus k \\
& y_{iS} \in \{0,1\} && \forall i \in N \setminus k, \forall S \subseteq M
\end{aligned}
$$

   This is the allocation problem assuming that player k does not participate in the auction.

4. Bundles are awarded to agents according to y^*. The payment that agent k pays is equal to

$$
V^{-k} - \left[V - \sum_{S \subseteq M} v_{kS} y^*_{kS} \right]. \tag{15.21}
$$

Note that the VCG auction awards bundles in the same manner as the combinatorial auction in Section 15.3.1, but the payments are different.

Here is the logic behind (15.21). V^{-k} is the "welfare" of the other agents when agent k is excluded from the auction. The term inside the brackets is the welfare of the other agents when agent k participates. So agent k's payment is equal to the difference in the other agents' welfare without him and with him. In other words, agent k reimburses the system for the value that he has taken away by winning his bundle.

Why does agent k have an incentive to tell the truth when he reports v_{kS}? First notice that changing v_{kS} doesn't affect V^{-k} since the V^{-k} auction excludes agent k. Moreover, the term inside the brackets in (15.21) is equal to the optimal objective value of the allocation problem minus agent k's payment. So that term is independent of v_{kS}. Therefore, a winning agent's payment will not change if he under- or over-states his valuation. Finally, if an agent over-states his valuations in the hope of winning a bundle that he wouldn't have won under truthful bidding, then he will pay more for this bundle than his valuation—see Problem 15.4. Therefore, agents have no incentive to misrepresent their valuations when they bid.

If the seller implements the VCG auction, her total revenue will be

$$\begin{aligned}\sum_{k\in N} V^{-k} - \sum_{k\in N}\left[V - \sum_{S\subseteq M} v_{kS}y^*_{kS}\right] &= \sum_{k\in N}\sum_{S\subseteq M} v_{kS}y^*_{kS} + \sum_{k\in N}(V^{-k} - V)\\ &= V + \sum_{k\in N}(V^{-k} - V). \qquad (15.22)\end{aligned}$$

If the number of bidders is large, then V will tend to be very close to V^{-k} since no single agent would have too large an effect on the auction. Therefore, the seller's revenue is close to V, which is the maximum revenue any auction could earn.

□ EXAMPLE 15.1

Suppose first that a single item—an antique book by Jane Austen—is being auctioned off. Three bidders are competing to buy the book. Their valuations, as reported in Step 1 of the VCG auction, are listed in Table 15.2. (The subscript A stands for "Austen.") The optimal allocation in Step 2 is to award the book to the highest bidder, bidder 2, for a total valuation of $V = 120$. In Step 3, the optimal allocation if bidders 1 or 3 are removed from the auction is still to award the book to bidder 2, with total valuation 120. If bidder 2 is removed, then the item goes to the second-highest bidder, bidder 1, for a valuation of 100. So the V^{-i} values are 120, 100, 120 (respectively). Bidder 2's payment is therefore $V^{-2} - [V - v_{2,A}y^*_{2,A}] = 100 - [120 - 120] = 100$. Note that 100 is also equal to the second-highest bid, confirming that the VCG auction operates as a sealed-bid second price auction in this case.

Now suppose that two books, one by Jane Austen and one by William Blake, are being auctioned off. The bidders' valuations for these items, and for the bundle consisting of both of them, are given in Table 15.3. (Subscript B is for "Blake.") The optimal allocation in Step 2 is to award the Austen book to bidder 2 and the Blake book to bidder 1, for a total valuation of $V = 220$. If bidder 1 is removed from the auction, the optimal allocation is to award Austen to 2 and Blake to 3, so $V^{-1} = 190$. If bidder 2 is removed from the auction, the optimal allocation is to

Table 15.2 Single-item VCG auction for Example 15.1.

Bidder (i)	$v_{i,A}$	$v_{i,A}y^*_{i,A}$	V^{-i}	Payment
1	100	0	120	0
2	120	120	100	100
3	10	0	120	0
		$V = 120$		100

award both books (the bundle AB) to bidder 1, so $V^{-2} = 200$. And if bidder 3 is removed, the optimal allocation is unchanged, so $V^{-3} = 220$. Therefore, bidder 1 pays $V^{-1} - [V - \sum_S v_{1S}y^*_{1S}] = 190 - [220 - 100] = 70$ for the Austen book and bidder 2 pays $V^{-2} - [V - \sum_S v_{2S}y^*_{2S}] = 200 - [220 - 120] = 100$ for the Blake book. The total payment is 170.

Table 15.3 Two-item VCG auction for Example 15.1.

Bidder (i)	$v_{i,A}$	$v_{i,B}$	$v_{i,AB}$	$\sum_S v_{iS}y^*_{iS}$	V^{-i}	Payment
1	100	100	200	100	190	70
2	120	0	0	120	200	100
3	10	70	120	0	220	0
				$V = 220$		170

□

It is well known that the VCG mechanism is strategy proof, *ex post* individually rational, and efficient. However, the VCG mechanism is not budget balanced; that is, it is possible that the auctioneer may receive a negative payoff. In a one-sided auction such as the one discussed here (with a single seller and multiple buyers), the budget imbalance arises from the fact that the auctioneer's valuations for the items are not considered in the allocation problem or payment calculation, and therefore the auctioneer could receive payments that fall short of her valuation for the items sold. (In Example 15.1, if the auctioneer's total valuation for the two books is 200, then she receives a negative payoff.) The same is true for VCG mechanisms in more complicated settings, such as double auctions (with multiple buyers and sellers).

Recently, there has been a focus on using auctions for supply chain procurement and trading in e-marketplaces. The benefits of auctions include lower information, transaction, and participation costs; increased convenience for both sellers and buyers; and, consequently, better market efficiency. While research and practice in operations management have emphasized optimizing the total supply chain, classical auction theory does not consider the operational costs associated with integrating a supply chain, such as logistics and inventory management costs. More recent work attempts to include these costs into the auction design. For example, Chen et al. (2005) consider combinatorial procurement auctions in supply chain settings. They incorporate supply chain costs (e.g., transportation costs in a complex supply chain network) into VCG auctions. Chu and Shen (2006, 2008) propose several double-auction mechanisms for e-marketplaces. Under their proposed double-auction mechanisms, bidding one's true valuation is the optimal strategy for each individual buyer and seller, even when shipping costs and sales taxes are different across various possible transactions. The proposed mechanism also achieves budget balance and asymptotic efficiency (that is, the auction approaches the maximum social welfare as the number of buyers and/or sellers approaches infinity). Furthermore, these results not only hold for an environment in which buyers and sellers exchange identical commodities, but also can be extended to more general environments, such as multiple substitutable commodities and bundles of commodities.

15.4.2 Weaknesses of the VCG Auction

Despite its impressive theoretical virtues of being strategy-proof, *ex post* individually rational and efficient, the VCG auction also suffers from several weaknesses that limit its usefulness in practice. In this section, we demonstrate some of these weaknesses using examples adapted from Ausubel and Milgrom (2006).

The first weakness of the VCG auction is that the auctioneer's revenues can be very low or zero, even when the items are valuable and the there are many competing bidders. We already know from Section 15.4.1 that the auctioneer's *payoff*—her revenue minus valuation—could be negative. Here we are arguing that her *revenue* could even be very small. The next example illustrates this.

□ **EXAMPLE 15.2**

Suppose three bidders are bidding for two parcels of land, denoted A and B. The bidders' valuations (in \$M) of the three possible bundles are given in Table 15.4. In Step 1 each bidder reports its values truthfully, and the optimal allocation in Step 2 is to award A to bidder 2 and B to bidder 3 (or, A to bidder 3 and B to bidder 2, which leads to a similar outcome). But the auctioneer's revenue is \$0, even though the total welfare of the bidders is maximized and the outcome is efficient. (In contrast, if the auction had been run as an English auction with a single item, the bundle AB, then all three bidders would have bid \$2 million, so the final price would have been \$2 million.)

Table 15.4 Valuations that induce zero revenue in the VCG auction.

Bidder (i)	$v_{i,A}$	$v_{i,B}$	$v_{i,AB}$	$\sum_S v_{i,S} y^*_{i,S}$	V^{-i}	Payment
1	0	0	2	0	4	0
2	2	2	2	2	2	0
3	2	2	2	2	2	0
				$V = 4$		0

□

Since the final auction revenue is one of the most important attributes for the auctioneer, the fact that the VCG auction can result in such small revenues for the auctioneer is a huge strike against it in practice.

Not only can the auctioneer's revenue be small, it is not, in general, monotonic in either the number of bidders or their valuations. We would expect that when the number of bidders increases, or their bids increase, the auctioneer's revenue should also increase, but this is not necessarily the case, as the next example demonstrates. Therefore, in the VCG auction, the auctioneer may prefer to prevent new bidders, or high-valuation bidders, from entering the auction, which is counterintuitive.

☐ **EXAMPLE 15.3**

Suppose that bidder 3 were to drop out of the auction described in Example 15.2, or equivalently, that his bid for bundle AB drops from 2 to 0; see Table 15.5. In this case, it is optimal to award both items to either bidder 1 or bidder 2, and in either case the auctioneer's revenue is \$2 million—an *increase* from Example 15.2, in which there were more bidders (or equivalently, higher bids).

Table 15.5 Valuations that induce nonmonotonicity in the VCG auction.

Bidder (i)	$v_{i,A}$	$v_{i,B}$	$v_{i,AB}$	$\sum_S v_{i,S} y^*_{i,S}$	V^{-i}	Payment
1	0	0	2	0	2	0
2	2	2	2	2	2	2
3	0	0	0	0	2	0
				$V = 2$		2

☐

Third, although we know from Section 15.4.1 that truthful bidding is a dominant strategy for each individual bidder in the VCG auction, it turns out that losing bidders can sometimes profit by colluding to deviate from their true valuations. In general, it is undesirable for an auction mechanism to incentivize collusion—another strike against VCG.

☐ **EXAMPLE 15.4**

Suppose that the valuations of bidder 1 are unchanged from Example 15.2, but the valuations of bidders 2 and 3 are reduced as shown in Table 15.6. If these bidders report their valuations truthfully, the VCG auction awards both items to bidder 1; bidders 2 and 3 lose the auction. However, suppose bidders 2 and 3 were to collude in advance to bid 2 for each item and for the bundle AB; then we have the auction in Example 15.2, for which we know that bidders 2 and 3 will each win one item, at a price of 0. Therefore, these bidders have an incentive to collude to misreport their true valuations.

Note that the collusion is essential here. For example, suppose bidder 2 continues to report her true valuation but bidder 3 misreports his valuations as 2 for every item and bundle; see Table 15.7. (Bidder 3 still values the items as in Table 15.6 but

Table 15.6 Valuations that induce collusion in the VCG auction.

Bidder (i)	$v_{i,A}$	$v_{i,B}$	$v_{i,AB}$	$\sum_S v_{i,S} y^*_{i,S}$	V^{-i}	Payment
1	0	0	2	2	1	1
2	0.5	0.5	1	0	2	0
3	0.5	0.5	1	0	2	0
				$V = 2$		1

reports them as in Table 15.7.) Then bidder 3 is awarded an individual item and pays 1.5 for it, even though his valuation for it is only 0.5.

Table 15.7 Noncollusive bids that fail to game the VCG auction.

Bidder (i)	$v_{i,A}$	$v_{i,B}$	$v_{i,AB}$	$\sum_S v_{i,S} y^*_{i,S}$	V^{-i}	Payment
1	0	0	2	0	2.5	0
2	0.5	0.5	1	0.5	2	0
3	2	2	2	2	2	1.5
				$V = 2$		1

□

A fourth weakness of the VCG auction is that it is vulnerable to bidders establishing multiple identities, i.e., a single bidder representing herself as multiple bidders. The next example demonstrates.

□ **EXAMPLE 15.5**

Suppose that there are only two bidders, 1 and 2, with the valuations given in Example 15.4; see Table 15.8. Then the VCG auction will award both items to bidder 1, at a price of 1; bidder 2 loses the auction. However, suppose bidder 2 creates a fake identity (e.g., a fake user account on the auctioneer's website) called "bidder 3" and submit the bids in Table 15.4 for both identities. From Example 15.2 we know that this will result in bidders 2 and 3 being awarded one item each but paying 0. Therefore, the "real" bidder 2 now becomes a winner; in fact, she will win both items, for free.

□

In summary, in addition to the computational complexity of solving the CAP, the examples above reveal important defects of the VCG auction—revenue deficiency, nonmonotonicity of the auctioneer's revenue, and a vulnerability to both collusion and multiple identities—that make it unappealing for most practical applications. For further discussion of these and other defects, see Ausubel and Milgrom (2006).

Table 15.8 Valuations that induce multiple identities in the VCG auction.

Bidder (i)	$v_{i,A}$	$v_{i,B}$	$v_{i,AB}$	$\sum_S v_{i,S}y^*_{i,S}$	V^{-i}	Payment
1	0	0	2	2	1	1
2	0.5	0.5	1	0	2	0
				$V = 2$		1

An important question is, under what conditions can these defects be eliminated? We will characterize one such condition from the perspective of game theory in the next section.

15.4.3 VCG Auction as a Cooperative Game

The examples in Section 15.4.2 demonstrate that the auctioneer's revenue in the VCG auction can be very low, even when the bidders' valuations for the items are high. A natural question is, how much revenue is acceptable for the auctioneer if we regard both the auctioneer and the bidders as players in a cooperative game? To answer this question, we will analyze the *core* of the game, which contains the payoff vectors from which no subset of the players—called a *coalition*—has an incentive to deviate. The core thus generalizes the concept of Nash equilibrium in a noncooperative game, which is a payoff vector from which no *single* agent has an incentive to deviate. Therefore, if the payoff vector (or allocation) of the VCG auction lies in the core, we consider the corresponding auctioneer's revenue to be acceptable.

In what follows, we first formulate the VCG auction as a cooperative game and show that the payoff vector is beneficial for the bidders. Next, we introduce a submodularity condition that guarantees that the VCG payoff vector lies in the core, and thus the whole coalition won't break. Finally, we briefly discuss a condition on the bidders' preferences that ensures that the submodularity condition holds, and therefore that the weaknesses discussed in Section 15.4.2 will never occur.

As usual, let N be the set of bidders and let M be the set of objects being auctioned off. Let 0 represent the auctioneer, and denote the set of all players as $L = \{0\} \cup N$. The *coalitional value function* is defined for coalitions $T \subseteq L$ as the maximum total value the coalition can create by trading among themselves, if the coalition includes the auctioneer; otherwise, the function equals zero. Here, we implicitly assume that the valuation of the auctioneer for any bundle is zero. Mathematically, for any $T \subseteq L$ such that $0 \in T$, the coalitional value function is

$$
\begin{aligned}
V(T) = \text{maximize} \quad & \sum_{i \in T \setminus 0} \sum_{S \subseteq M} v_{iS} y_{iS} \\
\text{subject to} \quad & \sum_{S \ni j} \sum_{i \in T \setminus 0} y_{iS} \leq 1 && \forall j \in M \\
& \sum_{S \subseteq M} y_{iS} \leq 1 && \forall i \in T \setminus 0 \\
& y_{iS} \in \{0, 1\} && \forall i \in T \setminus 0, \forall S \subseteq M,
\end{aligned}
$$

where, as usual, v_{iS} is the valuation of bidder i for bundle S. In other words, $V(T)$ is the optimal objective function value of the auctioneer's problem in the VCG auction, with N replaced by $T \setminus 0$.

The core of a game with player set L and coalitional value function $V(\cdot)$ is defined as follows:

$$C(L, V) = \left\{ \pi : V(L) = \sum_{i \in N} \pi_i, V(T) \leq \sum_{i \in T} \pi_i \ \forall T \subset L \right\}.$$

The core contains all payoff vectors π so that the coalitional value function of the whole player set equals the sum of the payoffs, and so that no smaller coalition could do any better than π if they excluded the other players. In other words, any payoff vector in the core is feasible, and the corresponding outcome is stable, in the sense that there is no coalition with the desire to change the outcome of the game. Thus, any payoff vector in the core is also referred to as a *competitive* outcome, for which it is impossible for the auctioneer and a subset of bidders to block the auction by defecting and negotiating an outcome with higher payoffs for themselves. Classical economic theory typically treats the core as the solution to a game-theoretic problem. From the perspective of game theory, a given type of auction is no more than a way to determine the payoff allocations in a game consisting of the auctioneer and bidders. In what follows, we discuss under what conditions the payoff vector of the VCG auction lies in the core. Our analysis is adapted from Ausubel and Milgrom (2006).

Let $\bar{\pi} = (\bar{\pi}_k)_{k \in L}$ be the VCG payoff vector. Specifically, $\bar{\pi}_k$ is the payoff of bidder $k \in N$:

$$\bar{\pi}_k = \sum_{S \subseteq M} v_{kS} y^*_{kS} - \left(V(L \setminus k) - \left[V(L) - \sum_{S \subseteq M} v_{kS} y^*_{kS} \right] \right) = V(L) - V(L \setminus k). \tag{15.23}$$

(Recall that the payoff is the valuation minus the price. The first term in (15.23) is the bidder's valuation, and the second is the price, from (15.21).) Similarly, $\bar{\pi}_0$ is the payoff of the auctioneer, 0:

$$\bar{\pi}_0 = V(L) + \sum_{k \in N} (V(L \setminus k) - V(L)) = V(L) - \sum_{k \in N} \bar{\pi}_k. \tag{15.24}$$

(See (15.22).)

The following Lemma shows that the core $C(L, V)$ is nonempty, and that every bidder's payoff in the VCG auction is the greatest among all points in the core of $C(L, V)$.

Lemma 15.1 *$C(L, V) \neq \emptyset$, and $\bar{\pi}_k = \max\{\pi_k : \pi \in C(L, V)\}$, for all $k \in N$.*

Proof. For any $k \in N$, consider a payoff vector π defined by $\pi_0 = V(L \setminus k)$, $\pi_k = V(L) - V(L \setminus k)$, and $\pi_l = 0$ for any $l \neq k$. From the definition of $V(\cdot)$, it is clear that $\pi \in C(L, V)$. Furthermore, we have $\bar{\pi}_k = \pi_k \leq \max\{\pi_k : \pi \in C(L, V)\}$.

Now suppose there exists a payoff vector $\pi \in C(L, V)$ such that $\pi_k > \bar{\pi}_k$ for some $k \in N$. Then $\sum_{l \neq k} \pi_l = V(L) - \pi_k < V(L) - \bar{\pi}_k = V(L \setminus k)$, which contradicts the fact that $\pi \in C(L, V)$. Hence, for any $\pi \in C(L, V)$ and $k \in N$, $\bar{\pi}_k \geq \pi_k$, which completes the proof. ∎

We say that a payoff vector π is *bidder dominant* if it lies in the core and if, for any $\pi' \in C(L, V)$, we have $\pi_k \geq \pi'_k$ for every bidder $k \in N$. In other words, π is bidder dominant if it is in the core and if it is at least as good as every other vector in the core, for every bidder. From Lemma 15.1, we have the following result.

Theorem 15.2 *If the VCG payoff vector $\bar{\pi}$ lies in the core, then it is the bidder-dominant point; otherwise, there is no bidder-dominant point in the core, and the auctioneer's VCG payoff is strictly less than the smallest of the auctioneer's core payoffs.*

Proof. The first part of the theorem follows from Lemma 15.1. Conversely, suppose $\bar{\pi} \notin C(L,V)$ and consider any $\hat{\pi} \in C(L,V)$. By Lemma 15.1, $\hat{\pi}_l \leq \bar{\pi}_l$ for any $l \in N$. Since $\hat{\pi} \neq \bar{\pi}$, there exists a bidder $k \in N$ such that $\hat{\pi}_k < \bar{\pi}_k$. So, $\hat{\pi}_0 = V(L) - \sum_{l \in N} \hat{\pi}_l > V(L) - \sum_{l \in N} \bar{\pi}_l = \bar{\pi}_0$. Consider the payoff vector $\pi \in C(L,V)$ defined by $\pi_0 = V(L \backslash k)$, $\pi_k = V(L) - V(L \setminus k) = \bar{\pi}_k$, and $\pi_l = 0$ for any $l \neq k$. We have $\hat{\pi}_k < \pi_k$, and thus $\hat{\pi}$ is not bidder dominant. ∎

Theorem 15.2 shows that the auctioneer's revenue from the VCG auction is strictly smaller than his revenue in any competitive outcome, unless the VCG payoff vector lies in the core.

Next we give conditions on the coalitional value function that ensure that the VCG payoff vector lies in the core, regardless of which potential bidders decide to participate in the auction. To that end, define $\bar{\pi}(S)$ as a payoff vector in the game in which the players are exactly the members of coalition S:

$$\begin{aligned} \bar{\pi}_k(S) &= V(S) - V(S \setminus k) \quad \text{for } k \in S \setminus 0 \\ \bar{\pi}_0(S) &= V(S) - \sum_{k \in S \setminus 0} \bar{\pi}_k(S). \end{aligned}$$

We say that the coalitional value function $V(\cdot)$ is *bidder-submodular* if, for all $k \in L \setminus 0$ and all coalitions S and S' such that $0 \in S \subset S'$,

$$V(S \cup \{k\}) - V(S) \geq V(S' \cup \{k\}) - V(S').$$

The following theorem shows that bidder-submodularity is a necessary and sufficient condition such that the VCG payoff vector lies in the core, and thus the auctioneer's VCG payoff meets the competitive benchmark.

Theorem 15.3 *The following three statements are equivalent:*

(i) *The coalitional value function $V(\cdot)$ is bidder-submodular.*

(ii) *For every coalition $S \ni 0$, $C(S,V) = \Pi_S$, where*

$$\Pi_S = \left\{ \pi_S : \sum_{k \in S} \pi_k = V(S), 0 \leq \pi_k \leq \bar{\pi}_k(S) \ \forall k \in S \setminus 0 \right\}.$$

(iii) *For every coalition $S \ni 0$, $\bar{\pi}(S) \in C(S,V)$.*

Proof. (i) $\Rightarrow$ (ii): From Lemma 15.1, $C(S,V) \subseteq \Pi_S$. Next we show that for any $\pi_S \in \Pi_S$, we have $\pi_S \in C(S,V)$, and hence $\Pi_S \subseteq C(S,V)$. To this end, we only need to verify the blocking inequalities, that is, for any $S' \subseteq S$, $\sum_{k \in S'} \pi_k \geq V(S')$. If

$0 \notin S'$, we have $\sum_{k \in S'} \pi_k \geq 0 = V(S')$; otherwise, without loss of generality, suppose $S = \{0, 1, \dots, s\}$ and $S' = \{0, 1, \dots, s'\}$ $(s \geq s')$. Then:

$$\begin{aligned}
\sum_{k \in S'} \pi_k &= V(S) - \sum_{l=s'+1}^{s} \pi_l \\
&\geq V(S) - \sum_{l=s'+1}^{s} \bar{\pi}_l(S) \\
&= V(S) - \sum_{l=s'+1}^{s} (V(S) - V(S \setminus l)) \\
&\geq V(S) - \sum_{l=s'+1}^{s} (V(\{0, 1, \dots, l\}) - V(\{0, 1, \dots, l-1\})) \\
&= V(S) - [V(S) - V(S')] \\
&= V(S'),
\end{aligned}$$

where the first inequality follows from $\pi_k \leq \bar{\pi}_k(S)$ and the second inequality follows from bidder-submodularity.

(ii) $\Rightarrow$ (iii) follows from the fact that $\sum_{k \in S} \bar{\pi}_k = V(S)$.

(iii) $\Rightarrow$ (i): Suppose V is not bidder-submodular. Then, there exists a player i such that $V(T) - V(T \setminus i)$ is not weakly decreasing in T. That is, there is a coalition S including the auctioneer and bidders $i, j \in S \setminus 0$ such that $V(S) - V(S \setminus i) > V(S \setminus j) - V(S \setminus \{i, j\})$. Hence,

$$\begin{aligned}
\sum_{l \in S \setminus \{i,j\}} \bar{\pi}_l(S) &= V(S) - \bar{\pi}_i(S) - \bar{\pi}_j(S) \\
&= V(S) - [V(S) - V(S \setminus i)] - [V(S) - V(S \setminus j)] \\
&= V(S \setminus i) + V(S \setminus j) - V(S) \\
&< V(S \setminus \{i, j\}).
\end{aligned}$$

Thus, the payoff vector $\bar{\pi}(S)$ is blocked by coalition $S \setminus \{i, j\}$. Therefore, there exists a coalition $S \ni 0$, such that $\bar{\pi}(S) \notin C(S, V)$, which contradicts (iii). ∎

Theorem 15.3 shows under what conditions on the coalitional value function the (restricted) VCG payoff vector lies in the core. Unfortunately, it is difficult for the auctioneer to know the coalition value function in advance. Therefore, it is worth investigating conditions under which the *individual* preferences guarantee that the VCG payoff vector lies in the core. See Ausubel and Milgrom (2006) for a further discussion of this issue.

CASE STUDY 15.1 Procurement Auctions for Mars

Mars, Inc. is a global manufacturer of candy, pet food, beverages, and other consumer goods, with over \$30 billion in sales in 2015. The food and pet food industries typically have lower profit margins than other consumer goods, meaning that low-cost procurement (purchasing) is critical for Mars's profitability. Mars decided to use *reverse auctions* to procure some of its raw materials rather than the more traditional one-on-one negotiations that Mars's buyers typically engaged in with its suppliers. In a reverse auction (also known as a procurement auction), there is one buyer and many suppliers, and the suppliers compete for the buyer's business. Mars partnered with experts at IBM T.J. Watson Research Center to develop the optimization models, algorithms, and web-based implementation of these auctions. Their project is detailed by Hohner et al. (2003).

Many of Mars's suppliers provide quantity discounts, like those discussed in Section 3.4, in order to encourage large purchases and to exploit the suppliers' economies of scale. In other cases, suppliers prefer to supply multiple items simultaneously—such as multiple sizes or colors of the same type of packaging. This allows them to take advantage of economies of scope—for example, by reducing setups on the production line that prints the packaging labels. Mars introduced two types of reverse auctions to accommodate these supplier preferences: supply-curve auctions (in which suppliers' bids take the form of supply curves, i.e., a set of quantity ranges and associated prices) and combinatorial auctions (similar to the mechanism discussed in Section 15.3). We discuss their combinatorial auction here; see Hohner et al. (2003) for details about the supply-curve auction.

The combinatorial auction described in Section 15.3 involves a single round: Bidders submit their bids, and the auctioneer allocates the items. However, it was important to Mars to have an iterative process consisting of multiple rounds; in each round, suppliers submit bids, Mars announces provisional winning bids, and the suppliers can revise their bids if they wish. This process has several advantages over the single-round approach: It avoids the need for suppliers to bid on all of the exponentially many possible bundles; it fosters increased competition among suppliers; and it allows suppliers to correct their bids based on updated information. Mars also imposed a number of side constraints on the auction, which constrain the number of winning suppliers as well as the quantity procured from each winning supplier.

Mars's combinatorial auction works as follows. (1) First, Mars posts a request for quote (RFQ) for a set M of items, as well as the quantity Q_j required of each $j \in M$. (2) Next, supplier $i \in N$ is allowed to submit up to K_i bids, where N is the set of suppliers. Supplier i's kth bid specifies a bundle $S \subseteq M$ and a price b_{ik} at which supplier $i \in N$ would be willing to supply all of each item contained in bundle S. We let $a_{ijk} = 1$ be a constant that equals 1 if supplier i's kth bid includes item $j \in M$, i.e., if item j is contained in the bundle specified in that bid. (3) Periodically, Mars solves the *winner-determination problem* (i.e., the auctioneer's problem, analogous to the CAP). The winner-determination problem uses two sets of decision variables: y_{ik} equals 1 if supplier i wins its kth bid, and 0 otherwise; and x_i equals 1 if supplier i wins *any* bid. The formulation is:

$$\text{minimize} \quad \sum_{i \in N} \sum_{k=1}^{K_i} b_{ik} y_{ik} \tag{15.25}$$

$$\text{subject to} \quad \sum_{i \in N} \sum_{k=1}^{K_i} a_{ijk} y_{ik} \geq 1 \qquad \forall j \in M \tag{15.26}$$

$$W_i^- x_i \leq \sum_{j \in M} \sum_{k=1}^{K_i} a_{ijk} Q_j y_{ik} \leq W_i^+ x_i \qquad \forall i \in N \tag{15.27}$$

$$Y^- \leq \sum_{i \in N} x_i \leq Y^+ \tag{15.28}$$

$$x_i \leq \sum_{k=1}^{K_i} y_{ik} \qquad \forall i \in N \tag{15.29}$$

$$x_i \in \{0, 1\} \qquad \forall i \in N \tag{15.30}$$

$$y_{ik} \in \{0, 1\} \qquad \forall i \in N, \forall k = 1, \ldots, K_i \tag{15.31}$$

The objective function (15.25) minimizes the total cost of the bids selected. Constraints (15.26) require every item to be contained in at least one winning bid (excess items are allowed), while (15.31) are integrality constraints. (So far, this formulation resembles a set covering problem and is similar to the formulation of the CAP except that it is indexed over the bids, rather than over the exponentially many bundles, since not all bundles will receive bids.) The remaining constraints are the side constraints: (15.27) ensures that the quantity awarded to each supplier i that wins a bid is between the constants W_i^- and W_i^+; (15.28) ensures that the total number of suppliers awarded bids is between the constants Y^- and Y^+; (15.29) ensures that x_i does not equal 1 if i is not awarded any bids; and (15.30) are integrality constraints on x. Mars solves the winner-determination problem using an off-the-shelf IP solver, which can efficiently solve problems with hundreds of items and thousands of bids. (4) Mars announces the provisional winning bundles and their associated prices. (5) Suppliers may use that information to revise their bids. (6) The process repeats until no suppliers wish to revise their bids, at which point the auction ends and the contracts are awarded. The whole process typically takes less than one hour.

Mars reported that the auctions led to considerable cost savings and that it recouped its investment in developing the auctions within a year. Moreover, Mars's savings did not come at the expense of suppliers' margins; in fact, supplier margins increased. The improved efficiencies in matching supply and demand led to this win–win result. This echoes the insight from Chapter 14 that improved supply chain coordination can lead to increased profits for both players—supply chain profits are not a zero-sum game.

PROBLEMS

15.1 (**Nonoptimality of the English Auction**) Suppose you have decided to sell a valuable collection of baseball paraphernalia. You have identified N potential buyers for the collection. In this problem, you will consider two alternate ways of selling the collection, one involving an auction and one not. You are selling the collection as a whole, not as individual parts.

Each of the N bidders has a valuation v_i for the collection. You do not know the v_is for each bidder, but you do know that each bidder's valuation is independently and uniformly distributed on $[0, 1]$.

Your first idea involves simply setting a price p and offering the collection for sale at that price. If some bidder wants to buy the item at that price (i.e., if $p < v_i$ for some bidder), he or she buys the item. If there are multiple such bidders, one is chosen randomly. If there are no such bidders, the sale ends unsuccessfully.

Table 15.9 Valuations for English auction in Problem 15.1.

Bidder (i)	Valuation (v_i)
1	100
2	120
3	135
4	85
5	90

a) Let $\gamma(p)$ be the probability that you sell the collection if you set the price to p. Calculate $\gamma(p)$. (Your answer should be in terms of N.)

b) Write your expected revenue as a function of p.

c) Calculate the optimal price p^*, the probability of selling the collection at this price, and the corresponding expected revenue.

d) Show that the optimal revenue is strictly increasing in N.

Next you consider selling the collection using an English auction, in which you start the price at $0 and gradually increase it until only one bidder remains. Assume that you increase the price continuously (infinitesimally), and that a bidder will not bid if the price equals his or her valuation (only if it's strictly less).

e) Argue that the English auction always results in the winning bidder paying the *second-highest* valuation.

f) The expected value of the second largest of N random variables that are iid $U[0,1]$ is equal to $(N-1)/(N+1)$. Use this fact to show that the expected revenue from the English auction is smaller than that from the nonauction method if $N = 1$ or 2 and is larger if $N \geq 3$. Therefore, the English auction is not optimal from the seller's perspective if $N < 3$.

g) Prove that, in the optimal solution to problem (D) on page 594 for this auction, p is equal to the second-highest valuation v_i.

h) Verify the result from part (g) by solving problem (LP) for the data in Table 15.9 using an LP solver of your choice. That is, verify that $x_i = 1$ for the bidder with the highest valuation but that p, the dual value for constraint (15.5), equals the second-highest valuation.

15.2 **(LP Relaxation of (CAP))** Construct a small example (using as few bidders and items as possible) for which the LP relaxation of (CAP) does not have an integer optimal solution.

15.3 **(VCG Auction for Candy Shipments)** The Truck o' Treats Company, a shipping company that specializes in refrigerated shipments of candy, will send a truck next week from Bethlehem, PA, to Chicago, IL, and from there to Berkeley, CA. The current load is insufficient to fill the truck, and the company plans to use a VCG auction to sell the remaining 1000 cubic feet of capacity to a candy company that needs to ship goods along those routes. (The remaining capacity is the same on both legs of the route because the truck will make a delivery in Chicago but pick up an equal volume of goods for shipment to Berkeley.)

Four candy companies each have 1000 cubic feet of candy to ship and are considering bidding for the routes. Horseshoe Candy needs to ship candy from Bethlehem to Chicago and is willing to pay up to \$900 for this leg; however, they have no product to ship to Berkeley and do not wish to bid for the second leg. Ares Chocolates, in contrast, has product to ship from Chicago to Berkeley (for which it is willing to pay \$1150) but nothing to ship from Bethlehem to Chicago. Valhalla Chocolates has goods to ship to both destinations; it is willing to pay \$600 to ship from Bethlehem to Chicago and \$1200 to ship from Chicago to Berkeley; however, if they can do both, they are willing to pay \$2000 to avoid the hassle of using two separate shipments. Similarly, W&W Candies is willing to pay \$800 for the Bethlehem–Chicago leg, \$950 for Chicago–Berkeley, and \$1900 for both.

What leg(s) will be awarded to each company in the outcome of the VCG auction, and what will each winning bidder pay? What will be the total revenue to Truck o' Treats? Construct a table like Table 15.3 that lists the bids, values of awarded items, V^{-i} values, and payments.

15.4 (Misrepresenting Valuations in the VCG Auction) Prove that, in the VCG auction, a bidder does not have an incentive to bid greater than his valuation (in an attempt to win a bundle that he otherwise would not have won). In particular, suppose that bidder k is not awarded a particular bundle $T \subseteq M$ if all bidders state their *true* valuations. Prove that if bidder k over-states his valuation for bundle T to such an extent that he is now awarded bundle T, then the price he pays for bundle T will be greater than or equal to v_{kT}, his true valuation for bundle T.

To keep things simple, you may assume that bidder k does not receive *any* bundle when all bidders state their true valuations, and that when bidder k over-states his valuation for T, the rest of the bundles are awarded to the same bidders that they were awarded to originally; that is, the allocation of bundle T may change, but no other bundles. Furthermore, assume the partitioning of items into bundles does not change.

Hint: First prove the result for a single-item, second-price auction.

15.5 (Double Auctions) A double auction consists of multiple buyers and multiple sellers. Potential buyers submit their bids and potential sellers simultaneously submit their ask prices to an auctioneer. The auctioneer first eliminates some sellers who ask too much and some buyers who offer too little, and then decides which of the remaining buyers and sellers will transact with each other, and at what price. Transactions incur costs, which may represent costs associated with transportation, quality, lead time, customization, and the buyer–vendor relationship. The transaction costs are assumed to be common knowledge.

Suppose there are multiple commodities to be exchanged in the auction. There is a collection of sellers, each of whom offers for sale a single unit of a single commodity, facing a collection of buyers, each interested in buying a bundle consisting of multiple commodities, but at most one of each. Formulate the problem of maximizing social welfare assuming all agents bid truthfully. Use the following notation:

I	set of buyers
J	set of sellers
C	set of indivisible commodities
f_i	bid price of buyer i for her bundle
g_j	ask price of seller j for his item
q_i	$= (q_i^c)_{c \in C}$, a bundle of goods that buyer i ($i \in I$) wants to procure; $q_i^c = 1$ if buyer i wants to procure one unit of commodity c and 0 otherwise

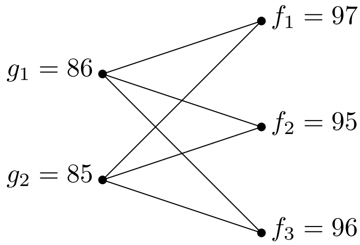


Figure 15.1 Bidders' valuations in double auction in Problem 15.6.

q_j $= (q_j^c)_{c \in C}$, supply offered by seller j ($j \in J$);
$q_j^c = 1$ if seller j supplies one unit of commodity c and 0 otherwise

d_{ijc} transaction cost when buyer i purchases commodity c from seller j

15.6 (Single-Commodity Double Auctions) Consider a simpler version of the double auction in Problem 15.5 in which there is only a single commodity. Assume that when buyer i trades with seller j, transaction cost d_{ij} is incurred.

a) Formulate the single-commodity double auction.

b) Show that the simplified formulation can be solved efficiently.

c) Consider an example with two sellers and three buyers. The transaction cost matrix is given in Table 15.10. The agents have an incentive to truthfully bid their valuations, which are shown in Figure 15.1. Determine which buyer(s) transact with which seller(s) in the efficient allocation. How much should the winning buyers pay? How much should the winning sellers receive?

Table 15.10 Transaction costs for double auction in Problem 15.6.

d_{ij}	1	2
1	4	7
2	6	4
3	9	6

15.7 (VCG Payoff Vector and Core) Consider a combinatorial auction with three items and four bidders. The bidders' valuations of the possible bundles are given in Table 15.11.

a) Suppose all bidders state their true valuations. Compute the auctioneer's revenue from the VCG auction, the bidders' payments, and the VCG payoff vector.

b) Does the VCG payoff vector lie in the core? If not, identify a coalition that will block this result.

Table 15.11 Valuations for VCG auction in Problem 15.7.

Bidder (i)	$v_{i,A}$	$v_{i,B}$	$v_{i,C}$	$v_{i,AB}$	$v_{i,AC}$	$v_{i,BC}$	$v_{i,ABC}$
1	4	2	2	4	4	3	8
2	3	5	2	3	5	5	8
3	4	2	6	4	5	3	7
4	2	2	2	3	4	4	10

CHAPTER 16

APPLICATIONS OF SUPPLY CHAIN THEORY

16.1 INTRODUCTION

Supply chain management is one of the domains in which the tools of operations research (OR) are most widely and successfully applied. But in the past few decades, the theory of supply chain management has matured to the point where it is now applied to other industries and application areas. That is, while supply chain theory is an application of the methodologies of OR, the methodologies of supply chain theory themselves are applied in many other domains. These include energy, health care, disaster relief, the environment, and nonprofit operations. In this chapter, we discuss some of the ways that the tools of supply chain theory—the tools that we have discussed in this book—have been applied to some of these areas. For additional discussion of how supply chain optimization, in particular, has been applied to energy, health care, and humanitarian relief, see Snyder (2017), Wu and Ouyang (2017), Zhao (2017), and Çelik et al. (2017).

16.2 ELECTRICITY SYSTEMS

Historically, electricity grids have functioned like the ultimate just-in-time supply chains, with no (or very little) inventory and almost instantaneous delivery of goods (i.e., energy). However, the modernization of electricity grids will provide new opportunities for optimizing their design and operation. Tomorrow's grids are likely to look a lot like today's supply chains, with inventories (in the form of large-scale batteries and other storage de-

Fundamentals of Supply Chain Theory, Second Edition. Lawrence V. Snyder and Zuo-Jun Max Shen.

Companion website: www.wiley.com/go/Snyder/SupplyChainTheory

vices), supply uncertainty (from volatile renewable generation sources such as wind and solar), demanding customer service requirements (as electricity markets continue to become deregulated and new competitors enter the marketplace), and novel pricing schemes (enabled by new communication infrastructure that can communicate pricing information in real time). In addition, classical principles of facility location will play a role in designing these grids, as will newer models for robust and resilient network design, as it becomes increasingly important to protect the grid from accidental or intentional disruptions that can affect the lives and livelihoods of millions of people. By viewing the grid as a supply chain network, we can leverage existing tools to develop a new generation of electricity systems.

Electricity grids are arguably the systems in which OR is most frequently used and plays the most critical role. Every 5–15 minutes, electricity system operators around the world solve an optimal power flow (OPF) problem to decide how much electricity each generator should produce for the next few minutes in order to meet the current demand and avoid overloading the power lines. The OPF problem is a nonlinear optimization problem (NLP), though it is often solved in a linearized form as a linear program (LP). And every day, these same operators solve the unit commitment (UC) problem, a thorny mixed-integer programming (MIP) problem, to determine which generators (units) should be operating during which hours in the next day. The Midwest Independent System Operator (MISO) won the prestigious INFORMS Edelman Award in 2011 for using UC problems to optimize energy markets (INFORMS 2011, Carlson et al. 2012). Many of these system operators also use optimization to solve auction problems, similar to those discussed in Chapter 15, to determine which generators should be used each day and at what rates they should be reimbursed.

However, since our focus is on applications of supply chain theory, and not of OR in general, we will not discuss these ubiquitous models; for reviews, see Frank et al. (2012a,b), Padhy (2004), and Ventosa et al. (2005), among others. Instead, we will discuss models for energy storage, transmission capacity planning, and power network design, which are built upon the tools of supply chain theory that are discussed in this book.

16.2.1 Energy Storage

Until the past few years, electricity could not be stored at a large scale—the supply of electricity had to balance the demand at all times. More recently, however, grid-scale energy storage has become technologically possible and financially practical. We can think of energy storage systems as large batteries, though other means for storing energy, such as flywheels, compressed air, and capacitors, are also being developed; see Akhil et al. (2013) for an in-depth discussion. Some of these methods (e.g., capacitors) store electrical energy, while others (e.g., batteries) convert electricity to chemical or other forms of energy and then convert it back when needed.

Energy storage is playing an increasingly important role in modern electricity grids. For example, although renewable energy such as wind and solar provides cheap, environmentally friendly power, it is also unpredictable—the energy production changes stochastically as the wind speed changes or the cloud cover moves. This poses problems for an electricity system operator, which needs predictability in order to manage the grid efficiently. Moreover, wind power tends to be highest at night, while the demand for electricity is highest during the day. System operators can alleviate both the unpredictability and the timing mismatch using energy storage: The storage system is charged when energy is plentiful and discharged when the energy is needed to meet the demand.

In the near future, electricity consumers may use energy storage in their own homes. Home energy storage systems can be charged at night (when electricity rates are often lower) and discharged during the day to power the home's appliances and devices. Homes that have solar panels or other forms of renewable energy can use storage to buffer against the uncertainty from these sources, as discussed above. And energy storage can also be used to buffer against uncertainty in the home's demand for electricity. These systems may be standalone storage systems, or even the batteries of plug-in hybrid electric vehicles (PHEVs), when the energy is not fully needed for travel. This is sometimes referred to as "behind-the-meter" energy storage since it operates on the consumer's side of the electricity meter, rather than on the power grid side.

One can think of an energy storage system as an inventory system in which the product being stored is energy, rather than some physical item. Therefore, many of the models for managing energy storage are based on the fundamental inventory optimization models discussed earlier in this book. On the other hand, energy storage systems usually have more decisions to make; for example, in addition to deciding how much "inventory" to buy (as in classical inventory models), we might also choose how much to sell and at what price. In this section, we discuss two inventory-like energy storage models: A behind-the-meter model for an electricity consumer and a model for a wind farm operator using storage to bid into an electricity market.

16.2.1.1 Behind-the-Meter Energy Storage Consider a large-scale battery located at a home (or an office building, university campus, etc.) that is capable of storing energy purchased from the grid. Energy that is stored in the battery can then be discharged, either to provide power for the devices in the home or to sell back to the grid. We will formulate an optimization model to decide how much energy to buy from and sell to the grid, and how much energy to charge and discharge the battery by, over a fixed planning horizon.

The amount of energy stored in the battery is called the *state of charge* and is expressed in units of kilowatt-hours (kWh). The battery has a fixed capacity (in kWh). It is also common to assume there is a limit on how much energy (kWh) can be charged to or discharged from the battery per unit time, but for the sake of simplicity, we will ignore this constraint. Another important aspect of energy storage models is the energy *losses* that result from charging or discharging the battery, or even when storing energy over time, due to inefficiencies in the storage and energy conversion processes. We will ignore this aspect of the model too, but it is not difficult to include it.

We will assume that the time horizon is divided into discrete periods. The demand for energy in each period is random, with a known probability distribution. (It is straightforward to modify this model to handle uncertainty in other aspects of the system, such as electricity prices.) Although energy is being bought, sold, and used continuously throughout each period, and the demand uncertainty is being revealed continuously, we will assume instead that everything happens in discrete periods, with the following sequence of events:

1. We decide how much energy to purchase from or sell to the grid.

2. We observe the random demand.

3. We charge/discharge the battery to make up any discrepancy between the energy purchased and the demand. If the energy purchased plus the battery state of charge is

insufficient to meet the demand, we incur an unmet-demand penalty. If the demand plus the remaining battery capacity is insufficient to absorb all of the energy purchased from the grid, the extra energy is lost; there is no explicit penalty, except that we pay for the energy even though we don't use it.

Note that other sequences of events are possible. For example, we might learn the demand *before* making buy/sell and charge/discharge decisions, in which case we have more information available when we optimize. In practice, the grid and battery decisions are made in nearly continuous time, so any discrete-time model is an approximation anyway. The sequence of events is a modeling decision and, as always, depends on a balance of realism and tractability.

Our goal is to maximize the expected profit (the expected revenue from selling energy minus the expected cost of purchasing it and the expected unmet-demand penalty) over the horizon. (We will actually formulate the problem as minimizing the negative of the profit.) We will use the following notation:

Parameters

T = number of time periods in horizon
c_t^+ = price of energy, per kWh, when buying from grid in period t
c_t^- = price of energy, per kWh, when selling to grid in period t
π_t = penalty per kWh of unmet demand in period t
B = capacity of battery, in kWh
x_0 = initial state of charge, in kWh, in period 1

Random Variable

D_t = demand for energy, in kWh, in period t; d_t is its realized value

Decision Variables

z_t = energy purchased from (> 0) or sold to (< 0) grid in period t
x_t = state of charge of battery at the *end* of period t

So, if $z_t > 0$, then we purchase z_t kWh from the grid in period t, and if $z_t < 0$, then we sell $-z_t$ kWh to the grid. In other words, z_t^+ is the energy purchased from the grid and z_t^- is the energy sold to it.

Once z_t has been chosen (step 1) and the actual demand d_t has been observed (step 2), the battery charge/discharge quantity and the possible unmet demand are fully determined. In particular, if we know x_t, the state of charge at the beginning of period t, then:

1. If $z_t \geq d_t$: We charge $\min\{z_t - d_t, B - x_t\}$ kWh to the battery; there is no unmet demand.

2. If $z_t < d_t$ and $z_t + x_t \geq d_t$: We discharge $d_t - z_t$ kWh from the battery; there is no unmet demand.

3. If $z_t + x_t < d_t$: We discharge x_t kWh from the battery; there is $d_t - (z_t + x_t)$ kWh of unmet demand.

We can combine these three cases to calculate the charge/discharge quantity and the unmet demand as follows:

$$\text{charge/discharge quantity} = \max\{\min\{z_t - d_t, B - x_t\}, -x_t\} \tag{16.1}$$

$$\text{unmet demand} = (d_t - (z_t + x_t))^+ . \tag{16.2}$$

(You should convince yourself that these equations correctly capture the logic described above.)

The problem of optimizing the buy/sell and charge/discharge decisions in this system and others like it can be modeled and solved as a multistage stochastic optimization problem; see, e.g., Korpaas et al. (2003), Castronuovo and Lopes (2004), and Brown et al. (2008). Alternately, one can use Markov decision processes (MDP), also known as stochastic dynamic programming; see, e.g., Kim and Powell (2011), Harsha and Dahleh (2015), and Zhou et al. (2018). This is the approach we will take, and you will find it very similar to the approach used in Section 4.3.3 for inventory systems.

Let $\theta_t(x)$ be the optimal total expected cost in period t through the end of the horizon, given that we start period t with a state of charge of x. Let $\theta_{T+1}(x)$ be the terminal cost function: If there are x kWh in the battery at the end of period T, we incur a cost of $\theta_{T+1}(x)$. Then $\theta_t(x)$ can be expressed recursively as follows:

$$\begin{aligned}\theta_t(x) = \min_z \Big\{ c_t^+ z^+ - c_t^- z^- + \mathbb{E}_{D_t}\left[\pi_t(D_t - (z-x))^+\right] \\ + \mathbb{E}_{D_t}\left[\theta_{t+1}\left(x + \max\{\min\{z - D_t, B - x\}, -x\}\right)\right]\Big\}. \quad (16.3)\end{aligned}$$

The first two terms inside the $\{\cdot\}$ are the grid purchase cost and sales revenue. The third term is the expected unmet-demand penalty, using (16.2). The fourth term is the expected cost for the rest of the horizon, given that we start with a state of charge of x and then charge/discharge by $\max\{\min\{z - D_t, B - x\}, -x\}$, from (16.1).

The recursion (16.3) is very similar to (4.36), except that (1) the decision variable in the minimization is z, the power bought or sold, rather than y, the new inventory level; (2) power may be bought *or* sold, so there is no constraint such as $y \geq x$, as in (4.36); and (3) the per-unit coefficient c depends on whether we buy ($z > 0$) or sell ($z < 0$) power. It can be solved very similarly, using Algorithm 4.1 with suitable modifications.

16.2.1.2 Energy Storage for a Wind Farm Operator Consider a wind farm operator that wishes to sell power to the electricity grid. Assume that there are two markets for buying and selling energy, a *day-ahead market* and a *real-time market*. (Many electricity systems around the world work this way.) In the day-ahead market, firms bid by indicating how much electricity they are willing to provide during each time interval (say, each hour) in the next day, and at what prices. The system operator then determines which bids to accept, typically by solving a UC problem, and the firms whose bids are accepted are obligated to provide the electricity they committed to, at prices that are determined by the system operator based on the bids. In contrast, the real-time market has no advance commitments; firms buy and sell electricity based on current prices, which change throughout the day.

Since wind power producers are typically small relative to the system as a whole, it is reasonable to assume that the wind farm operator has no control over the prices (the price component of its bid is irrelevant). Moreover, since wind is among the cheapest forms of electricity, we can also assume that bids from wind producers are always accepted. We will also make the simplifying assumption that tomorrow's day-ahead prices are known deterministically.

Our wind farm operator wishes to decide how much to bid into the day-ahead market for each hour tomorrow. The difficult issue here is that the wind power is stochastic, and moreover, if the firm cannot meet its bid during a given hour (because the actual wind power is insufficient), it must make up the difference in some other way, typically by buying power on the real-time market. Real-time prices are often lower than day-ahead prices—so the firm may actually come out ahead in this case—but they sometimes spike *much* higher than day-ahead prices, in which case the firm incurs significant costs. This problem is particularly acute for renewable energy producers, since the output of conventional energy generators can be controlled, while that of renewables is stochastic and largely out of the control of the operator.

In the models below, we will assume that the firm cannot sell to the real-time market, though this assumption can be relaxed (see Problems 16.3 and 16.4).

Suppose first that the wind farm has no energy storage available. Therefore, the 24 hours in tomorrow's day-ahead market are decoupled from each other, and we can solve each individually. Let Y be the wind power produced in a given hour; Y is a random variable with pdf $f_Y(\cdot)$ and mean μ_Y. Let c be the day-ahead price for that hour, which we assume is deterministic, and let R be the real-time price, which we assume is a random variable with pdf $f_R(\cdot)$ and mean μ_R. Our goal is to choose a value of Q, the bid quantity.

We can write the expected cost function as follows:

$$\begin{aligned} g(Q) &= -cQ + \mathbb{E}_{Y,R}[R(Q-Y)^+] \\ &= -cQ + \mu_R \mathbb{E}_Y[(Q-Y)^+]. \end{aligned} \tag{16.4}$$

The first term represents the revenue, which we treat as a negative cost, while the second represents the expected cost of buying power from the real-time market to make up any shortfall between the bid and the wind power produced. The second equality follows from the fact that Y and R are independent. The cost function in (16.4) is equivalent to the newsvendor cost function, plus a constant, and so can be solved easily; see Problem 16.2.

Now suppose that the wind farm has an energy storage system similar to that discussed in Section 16.2.1.1. The time periods are now coupled and must be solved simultaneously. We'll again use DP for this purpose, adding subscripts t to the notation defined above. Let $\theta_t(x)$ be the optimal total expected cost in period t through the end of the horizon, given that we start period t with a state of charge of x, and let $\theta_{T+1}(x)$ be the terminal cost function.

As in Section 16.2.1.1, once we choose the bid Q_t and observe the wind power y_t in period t, the battery charge/discharge quantities are determined, as is the amount of electricity that must be purchased on the real-time market. In particular, if x_t is the state of charge at the beginning of period t, then:

1. If $y_t \ge Q_t$: We charge $\min\{y_t - Q_t, B - x_t\}$ kWh to the battery and purchase no electricity from the real-time market.

2. If $y_t < Q_t$ and $x_t + y_t \ge Q_t$: We discharge $Q_t - y_t$ kWh from the battery and purchase no electricity from the real-time market.

3. If $x_t + y_t < Q_t$: We discharge x_t kWh from the battery and purchase $Q_t - (x_t + y_t)$ kWh from the real-time market.

Therefore, the charge/discharge quantity and the quantity purchased from the real-time market are as follows:

$$\text{charge/discharge quantity} = \max\{\min\{y_t - Q_t, B - x_t\}, -x_t\} \tag{16.5}$$

$$\text{real-time market quantity} = (Q_t - (x_t + y_t))^+ . \tag{16.6}$$

Then the recursion for $\theta_t(x)$ is given by:

$$\theta_t(x) = \min_{Q \geq 0} \left\{ -c_t Q + \mathbb{E}_{Y_t} \left[\mu_{R,t} (Q - (x + Y_t))^+ \right] \right.$$
$$\left. + \mathbb{E}_{Y_t} \left[\theta_{t+1} \left(x + \max\{\min\{Y_t - Q, B - x\}, -x\} \right) \right] \right\}. \quad (16.7)$$

The first term inside the $\{\cdot\}$ is the revenue from selling power on the day-ahead market, while the second is the expected cost of buying power on the real-time market, from (16.6). The third term is the expected cost for the rest of the horizon, given that we start with a state of charge of x and then charge/discharge by $\max\{\min\{Y_t - Q, B - x\}, -x\}$, from (16.5). Once again, the recursion is very similar to the inventory recursion (4.36) and can be solved using a modification of Algorithm 4.1.

16.2.2 Transmission Capacity Planning

Wind farms are often located in geographical areas with low population density, far from major load (demand) centers, and cannot be directly integrated into the existing electricity transmission network. As a result, long-distance transmission lines must be constructed to deliver electrical power from remote wind farms. In this section, we discuss a model for optimizing the capacity of such a transmission line. Our model is based on the work of Qi et al. (2015), though they model and solve the problem directly, whereas we will treat it as a newsvendor problem.

Suppose that a planner wishes to decide the capacity of the transmission line that connects a remote wind farm to the existing power grid. The system operator has implemented a *feed-in tariff* policy, a multiyear contract in which the wind farm is paid c per kWh transmitted to the grid, regardless of the quantity. Thus, the price is independent of quantity, as in Section 16.2.1.2, but unlike in that section, here the wind farm operator does not need to commit to a generation quantity in advance.

The wind farm is equipped with an energy storage system (battery) like that discussed in Section 16.2.1.2. We assume that the battery has effectively infinite capacity. Because of this, and because the feed-in tariff means there is no incentive to use storage to take advantage of price changes over time, we can ignore many of the details we modeled in Section 16.2.1.2.

Intuitively, the trade-off faced by the transmission line planner is that as the line capacity increases, so does the construction cost, but the energy losses decrease. The energy loss is due to the dissipation in energy due to friction as the battery is charged and discharged. (We ignored these energy losses in earlier sections.)

To quantify this trade-off, denote the transmission line capacity S kW. The annualized cost to build the transmission line is linear in the capacity, given by KS, where $K > 0$ is the cost per kW of capacity.[1] Let the charge and discharge efficiency of the battery be $\alpha, \beta \in (0, 1)$, respectively. In other words, if 1 kWh of energy is charged to the battery, the battery actually "receives" α kWh, and if 1 kWh is discharged from the battery, the transmission line actually "receives" β kWh from the battery. Thus, if we charge 1 kWh and subsequently discharge it, the net energy received from this "round trip" is $\alpha\beta$. The

[1]Generation and storage are measured in units of kWh, while transmission capacity is measured in kW. You can think of the line as transmitting S kWh per hour, i.e., S kW.

$1 - \alpha\beta$ kWh of energy that are lost to friction represent lost revenue. A larger transmission line capacity might have avoided this lost revenue, but would of course have cost more.

We can model this decision as a newsvendor problem. As in Section 16.2.1.2, let Y be a random variable that equals the wind power generated in one hour. For each kWh of difference between the wind power generated, Y, and the line capacity, S, we incur either a holding (overage) or stockout (underage) cost.

In particular, suppose the capacity is greater than the wind power in a given hour ($S > Y$). If we had built one fewer kW of capacity, we would have saved $K\delta$ in this hour, where $\delta = 1/(24 \cdot 365)$ converts the construction cost from years to hours. Therefore, the holding cost, the cost of building too much transmission capacity, is $h = K\delta$.

Now suppose instead that the capacity falls short of the wind energy ($S < Y$). The surplus energy is charged into the battery and then discharged at some future time. Each kWh of surplus energy results in an energy loss of $1-\alpha\beta$ and a revenue loss of $c(1-\alpha\beta)$. On the other hand, each kWh of surplus energy saved us $K\delta$ in transmission line construction costs for this hour. Therefore, the stockout cost, the cost of building too little transmission capacity, is $p = c(1 - \alpha\beta) - K\delta$.

Let $F(\cdot)$ be the cdf of Y, and let μ and σ^2 be its mean and variance. Then, from (4.16), the optimal capacity, S^*, solves

$$F(S) = \frac{p}{p+h} = \frac{c(1-\alpha\beta) - K\delta}{c(1-\alpha\beta)}. \tag{16.8}$$

If Y has a normal distribution, we can use (4.24) to get an explicit expression for S^*. However, wind power is often modeled using the Weibull distribution, typically with parameters that do not follow $\mu \gg \sigma$ and therefore are not conducive to approximating it with the normal distribution.

Instead, we can approximate the wind power distribution as $Y \sim U\left[\mu - \frac{\sigma^2}{2}, \mu + \frac{\sigma^2}{2}\right]$, which has been shown numerically to provide a good fit (Qi et al. 2015). The cdf of a $U[a, b]$ random variable is $F(x) = (x - a)/(b - a)$. Therefore, (16.8) becomes

$$\begin{aligned}
\frac{S - \left(\mu - \frac{\sigma^2}{2}\right)}{\sigma^2} &= \frac{c(1-\alpha\beta) - K\delta}{c(1-\alpha\beta)} \\
\iff \quad \frac{S-\mu}{\sigma^2} + \frac{1}{2} &= 1 - \frac{K\delta}{c(1-\alpha\beta)} \\
\iff \quad S &= \frac{\sigma^2}{2} + \mu - \frac{\sigma^2 K\delta}{c(1-\alpha\beta)}.
\end{aligned}$$

However, the transmission line capacity must be greater than or equal to the mean wind power, μ, otherwise the battery storage level will increase to infinity over time. Therefore, we have

$$\begin{aligned}
S^* &= \max\left\{\mu, \frac{\sigma^2}{2} + \mu - \frac{\sigma^2 K\delta}{c(1-\alpha\beta)}\right\} \\
&= \begin{cases} \mu + \left(\frac{1}{2} - \theta\right)\sigma^2, & \text{if } \theta < \frac{1}{2} \\ \mu, & \text{otherwise,} \end{cases}
\end{aligned}$$

where

$$\theta = \frac{K\delta}{c(1-\alpha\beta)}.$$

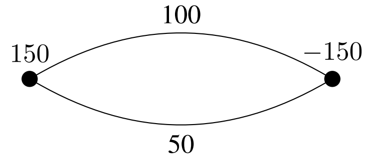


Figure 16.1 Two arcs with different capacities.

The constant θ captures the trade-off between the transmission capacity and the storage efficiency. If K is large—say, because the transmission line is long, or because construction is very expensive—or if the battery is efficient so that $\alpha\beta \approx 1$, then θ will be close to 1, and it will not be cost-effective to invest in extra transmission capacity (over and above μ).

16.2.3 Electricity Network Design

Electricity networks are fundamentally different from supply chain networks. In supply chain or other transportation networks, we can decide how much flow to send on each arc in the network (subject to capacity constraints). However, in electricity networks, it is the laws of physics, rather than our own decisions, that dictate these flows. We can decide how much electricity to inject into the network at generator nodes and how much to withdraw from the network at demand nodes, but after those decisions are made, the flows are determined by Kirchhoff's laws. One way to think about this difference is that in supply chain networks, we can control quantities on arcs, whereas in electricity networks, we can only control quantities at nodes. This makes designing and operating electricity networks more mathematically complex than doing the same for supply chain networks.

For a simple example, see the network in Figure 16.1. The node on the left supplies 150 units, while the node on the right demands 150 units. The upper and lower arcs have a capacities of 100 and 50, respectively, but are identical in all other respects. In a supply chain network, we can simply ship 100 units on the upper arc (road) and 50 on the lower. But in an electricity network, since the two arcs (power lines) have identical characteristics, 75 units will flow on each line, violating the capacity of the lower one.

In this section, we discuss a model for constructing a new electricity network. The model is based on the arc design model in Section 8.7.2 but accounts for the electricity flows. Many electricity network design models assume that a portion of the network has already been constructed, and we are deciding which additional links to add—they are network expansion rather than network design models—but we will omit this extra complication and assume that we are designing the network from scratch. It is straightforward to modify the model to account for existing links.

The terminology of electricity grid analysis differs somewhat from that of supply chain networks. Nodes are known as *buses*, and arcs or edges are known as *lines*. Let N be the set of buses and E be the set of potential lines. Demands are called *loads*, and demand buses are *load buses*. Each bus i has a *voltage*. Voltages are represented by complex numbers, and the relationship between power flows and voltages is nonlinear and nonconvex. However, we will instead consider the so-called *linearized* or *DC power flow* model, which ignores the imaginary component of the complex voltage and focuses instead on the so-called *real power*.

Bus i is assigned a *voltage angle* θ_i. The line between buses i and j (called line (i,j)) has a *susceptance* b_{ij}; very roughly speaking, when the susceptance is larger, power flows more easily through the line. In particular, the power p_{ij} that flows on line (i,j) is given by

$$p_{ij} = b_{ij}(\theta_i - \theta_j). \tag{16.9}$$

If $p_{ij} > 0$, then power is flowing from i to j, and vice versa. Bus i's total power is given by

$$p_i = \sum_{\substack{j\in N:\\(i,j)\in E}} p_{ij}; \tag{16.10}$$

if $p_i > 0$, then power is flowing out of bus i, and vice versa. The power at bus i is bounded above and below:

$$\underline{p}_i \le p_i \le \overline{p}_i; \tag{16.11}$$

if i is a load bus, then $\underline{p}_i = \overline{p}_i$ (and both are negative). Finally, the power flowing on line (i,j) is bounded by the line capacity, denoted $\overline{s}_{ij}$:

$$p_{ij} \le \overline{s}_{ij}. \tag{16.12}$$

Constraints (16.9)–(16.12) constitute the DC power flow model. The OPF problem discussed above has an objective function of minimizing the total generation cost and has (16.9)–(16.12) as constraints. But our interest is in using (16.9)–(16.12) to model the power flows in the network design problem.

To that end, let f_{ij} be the fixed cost to construct line (i,j), and let $x_{ij} = 1$ if we construct the line and 0 otherwise. The other decision variables in the model are p_{ij}, p_i, and θ_i. Then the electricity network design problem is:

$$\begin{aligned}
\text{minimize} \quad & \sum_{(i,j)\in E} f_{ij}x_{ij} && && (16.13)\\
\text{subject to} \quad & |p_{ij} - b_{ij}(\theta_i - \theta_j)| \le M(1-x_{ij}) && \forall (i,j)\in E && (16.14)\\
& \sum_{\substack{j\in N:\\(i,j)\in E}} p_{ij} = p_i && \forall i \in N && (16.15)\\
& \underline{p}_i \le p_i \le \overline{p}_i && \forall i \in N && (16.16)\\
& p_{ij} \le \overline{s}_{ij}x_{ij} && \forall (i,j)\in E && (16.17)\\
& x_{ij} \in \{0,1\} && \forall (i,j)\in E && (16.18)
\end{aligned}$$

Constraints (16.14)–(16.17) are the power flow constraints, modified to account for the construction variables x_{ij}. In (16.14), M is a large constant. If $x_{ij} = 1$, then we construct line (i,j), and (16.14) is equivalent to (16.9); otherwise, the constraint has no effect. This constraint is nonlinear because of the absolute value, but it can be linearized using standard methods. Similarly, if $x_{ij} = 1$, then (16.17) bounds the power flow on line (i,j) by the capacity, whereas if $x_{ij} = 0$, then p_{ij} is forced equal to 0. Finally, constraints (16.18) are integrality constraints. Note that p_{ij} is unrestricted in sign.

Electricity network design problems can be solved using various MIP optimization techniques such as Benders decomposition (Oliveira et al. 1995, Binato et al. 2001), dynamic programming (Dusonchet and El-Abiad 1973), or metaheuristics (Romero et al. 1995), or using off-the-shelf solvers (Alguacil et al. 2003).

16.3 HEALTH CARE

The United States spends over \$3 trillion on health care per year, representing over 17% of gross domestic product (GDP) (Hartman et al. 2018), and health care costs continue to grow. The health care system encompasses many complex supply chains, and there are many opportunities to improve the operations of these supply chains using the tools discussed in this book. Moreover, there are "virtual" supply chains within the health care system—flows of people, expertise, money, and other resources whose behavior can be modeled using many of the same techniques. In addition, the health care system consists of a huge number of individual parties—hospitals, doctors, insurers, pharmaceutical and device companies, patients—with often conflicting objectives. Coordination models of the type covered in this book will be useful tools for ensuring that the net result of the interactions among these parties is beneficial to patients and to society as a whole.

It has been estimated that between \$0.30 and \$0.40 of every dollar spent on health care—more than half a trillion dollars per year—is due to "overuse, underuse, misuse, duplication, system failures, unnecessary repetition, poor communication, and inefficiency" (Lawrence 2005). In 2005, the U.S. National Academy of Engineering and Institute of Medicine (now called the National Academy of Medicine) issued a report calling for the use of systems engineering, operations research, industrial engineering, and related engineering fields in health care delivery systems (National Academy of Engineering (US) and Institute of Medicine (US) Committee on Engineering and the Health Care System 2005). As the report notes:

> The experiences of other major manufacturing and services industries, which have relied heavily on systems-engineering concepts and tools to understand, control/manage, and optimize the performance of complex production/distribution systems to meet quality, cost, safety, and other objectives, can provide valuable lessons for health care.

In this section, we discuss two areas of health care in which the tools of supply chain theory specifically have been applied: production planning and contracting for influenza vaccines and inventory management for blood platelets.

16.3.1 Production Planning and Contracting for Influenza Vaccines

Influenza (flu) is a respiratory illness that spreads easily from person to person. Each year, millions of people worldwide get seriously ill with influenza, and hundreds of thousands die from the disease (World Health Organization 2018). Vaccination is the most effective means for controlling the disease.

Influenza vaccines can be thought of as newsvendor-type products, with a single selling season. This is because the strains of the influenza virus that are included in the vaccine change each year, based on analysis by the World Health Organization (WHO) of the virus strains that are most prevalent at the time. Therefore, last year's vaccines cannot be used this year. Vaccines that are administered year-round without seasonal effects, such as childhood vaccines for polio, measles, and the like, are also perishable (they have expiration dates). But their compositions do not change annually, and their shelf lives are typically longer than their ordering cycles. Therefore, their inventories can be managed using modified EOQ-type models.

On the other hand, the influenza vaccine has two key differences from the classical newsvendor assumptions: yield uncertainty and nonlinear sale value. Yield uncertainty

(Section 9.3) arises from the fact that the yield attained from the manufacturing process—which takes place largely inside embryonated chicken eggs—is difficult to predict and depends significantly on the virus strains that are included in the vaccine. The nonlinear sale value is a result of the epidemiology of the flu, namely, that the incremental benefit of one additional person vaccinated depends on the number of people who have already been vaccinated—it is not a constant.

Chick et al. (2008) introduce a model for the Stackelberg game that occurs when a large buyer (e.g., a government) purchases vaccines from a supplier. We present a simplified version of their model here.

In this Stackelberg game, the buyer (the government) is the leader and the supplier is the follower. (Note that this is the opposite of the sequence considered in the Stackelberg models in Chapter 14.) The government wishes to purchase enough vaccine to vaccinate a fraction f of its population of N individuals; the government's decision variable is f. Each individual to be vaccinated requires δ doses of the vaccine. Therefore, the government's total demand is $fN\delta$.

The supplier then decides how many eggs to inject with the virus, Q. The number of vaccine doses produced is given by the multiplicative yield QZ, where Z is a random variable with pdf f_Z. Each egg injected costs the supplier c_s. The government pays w to the supplier for each dose it purchases, and it also incurs a cost of c_g per individual vaccinated. Finally, there is a cost of b per infected individual, which can include treatment costs, lost wages, etc. Although b is a societal cost, we will assume it is borne by the government.

If we vaccinate a fraction f of the N individuals in the population, then the function $T(f)$ measures the expected number of individuals that will have been infected by the end of the influenza season. $T(f)$ can be calculated using epidemiological models; in this case, we use a standard model called the SIR model, where the acronym stands for the three "compartments" of individuals: susceptible, infectious, and recovered (Anderson and May 1992). Susceptible individuals are vulnerable to becoming infected with the disease; infectious individuals have the disease and can communicate it to others; and recovered individuals are no longer infectious (or were immune to begin with). Under standard assumptions, we get that $T(f) = pN$, where p, the "attack rate," satisfies

$$p = S(0)\left(1 + \frac{S(0)}{I(0)} - e^{-R_0 p}\right), \tag{16.19}$$

$S(0)$ and $I(0)$ are the initial numbers of susceptible and infectious individuals at the start of the season, and R_0 is the "basic reproduction number," i.e., the number of new cases generated by each new case, on average. For influenza, R_0 is roughly 2–3.

The function $T(f)$ is plotted in Figure 16.2. Clearly, it is nonlinear. This function indicates the value that the government gets as a function of its decision variable, which is why the problem has a nonlinear sale value, unlike the classical newsvendor problem. The nonlinearity itself is not necessarily a major impediment for the newsvendor problem, but the fact that p must be characterized implicitly (it appears on both the left- and right-hand sides of (16.19)) is.

On the other hand, $T(f)$ can be approximated well by a simple piecewise-linear function, also plotted in Figure 16.2. The breakpoint occurs at $f = f^0$, where f^0 is the "critical vaccination fraction," i.e., the smallest value of f that brings R_0 down to 1. The value of f^0 is straightforward to calculate in the SIR model.

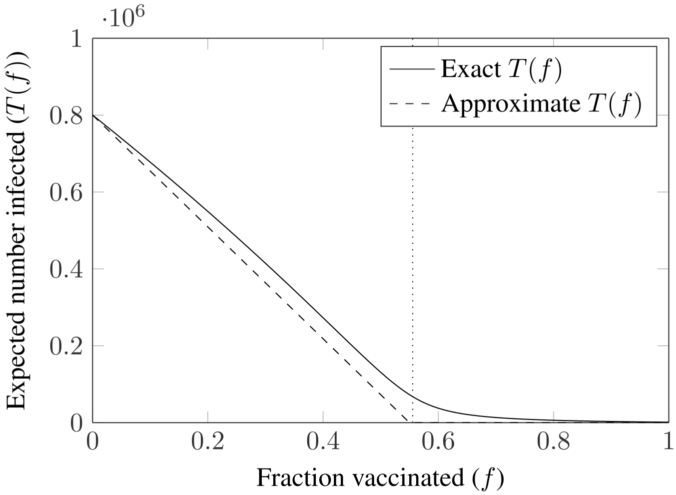


Figure 16.2 Exact and approximate $T(f)$ function for epidemiological model using $R_0 = 2.0$ and $N = 10^6$, with $S(0) = 0.99(1 - 0.9f)$ and $I(0) = 0.01(1 - 0.9f)$. Vertical dotted line indicates $f^0 = 0.56$.

Let's first look at the supplier's problem. Since the government acts first, the supplier already knows that the government's demand is $fN\delta$. It faces multiplicative yield uncertainty in a newsvendor-type setting. The actual number of doses produced, QZ, may be greater than, less than, or equal to $fN\delta$. Each dose of vaccine that is produced but not used incurs a cost of $h = c_s$, whereas each unit of unmet demand incurs a cost of $p = w - c_s$. (This is the supplier's stockout cost—the government's, and the population's, cost of vaccine shortages is quite different.)

The supplier's expected cost function is given by

$$g_s(Q, f) = c_s Q - w\mathbb{E}\left[\min\{QZ, fN\delta\}\right]. \tag{16.20}$$

The first term represents the cost of injecting Q eggs. The second represents the sales revenue, since the supplier sells the government either the number of doses produced or the government's order quantity, whichever is smaller. One can rewrite $g_s(Q, f)$ as

$$\begin{aligned} g_s(Q, f) = c_s(1 - \mathbb{E}[Z])Q &- (w - c_s)fN\delta \\ &+ c_s\mathbb{E}\left[(QZ - fN\delta)^+\right] + (w - c_s)\mathbb{E}\left[(fN\delta - QZ)^+\right]. \end{aligned} \tag{16.21}$$

(See Problem 16.5.) Compare (16.21) to the single-period newsvendor problem with deterministic demand and multiplicative yield uncertainty in Section 9.3.2.3. The last two terms are equivalent to (9.30) with $h = c_s$ (because each dose of the vaccine that is produced but not used incurs a cost of c_s), $p = w - c_s$ (for each unit of unmet demand, the supplier incurs a lost profit of $w - c_s$), and $d = fN\delta$. However, by setting $h = c_s$, we are implicitly assuming that we only pay for units that are produced. But the supplier in the vaccine model pays for every egg used, whether or not it yields a dose of vaccine. Therefore, we must add the expected cost of the eggs that are injected but do not yield vaccine doses; this is the first term in (16.21). Finally, the second term is similar to the additive constant that converts between the explicit and implicit versions of the newsvendor cost function; see Section 4.3.2.4 and Problem 4.15.

From (9.31), we get

$$\begin{aligned}\frac{dg_s(Q,f)}{dQ} &= c_s(1-\mathbb{E}[Z]) + c_s\mathbb{E}[Z] - w\int_0^{fN\delta/Q} zf_Z(z)dz \\ &= c_s - w\int_0^{fN\delta/Q} zf_Z(z)dz.\end{aligned}$$

Therefore, for a given f, the optimal quantity $Q(f)$ satisfies

$$\int_0^{fN\delta/Q(f)} zf_Z(z)dz = \frac{c_s\mathbb{E}[Z]}{w}.$$

Now consider the government's problem. The government wishes to choose f to minimize the total expected cost of infections and of purchasing and administering vaccines:

$$g_g(Q,f) = \mathbb{E}\left[bT\left(\frac{V}{N\delta}\right) + wU + c_gV\right], \tag{16.22}$$

where

$$\begin{aligned}U &= \min\{QZ, fN\delta\} \\ V &= \min\{QZ, fN\delta, f^0N\delta\}.\end{aligned}$$

U is the number of vaccine doses purchased: the minimum of the production yield and the number ordered. V is the number of doses administered: If the number purchased is less than $f^0N\delta$, we only administer $f^0N\delta$, because administering more doses is not cost-effective. (We omit the justification for this latter claim; see Chick et al. (2008).) The three terms in (16.22) represent the expected costs of infections, vaccine purchases, and vaccine administration, respectively. When the government chooses f, it knows that the supplier will choose $Q(f)$, so it can replace Q with $Q(f)$ in (16.22).

The arrangement described above is a wholesale price contract. The supplier's optimal production quantity, Q_s^*, under this contract is smaller than the system-optimal quantity Q^0, i.e., the quantity that a centralized decision-maker would choose in order to minimize the total cost system-wide cost. Therefore, the supply chain is not coordinated. The under-production occurs because the supplier bears all of the risk of arising from the yield uncertainty.

Chick et al. (2008) introduce a contract called a *cost-sharing contract* in which the government pays the supplier an additional cost per egg injected, in addition to the cost that it pays per vaccine dose delivered. The government therefore assumes some of the risk of excess production, inducing the supplier to increase its production quantity. If the contract parameters are set appropriately, the new production quantity is system-optimal, and the supply chain is coordinated.

16.3.2 Inventory Management for Blood Platelets

Blood platelets are used during many types of surgeries and are part of the treatment regimen for patients with leukemia and other cancers, as well as for a wide range of other medical conditions. The management of the inventory of platelets is particularly difficult because (1) platelets are usually in short supply, since donating platelets is much more time-consuming than regular blood donations; (2) demand for platelets is random and nonstationary, e.g., with less demand on weekends; and, most importantly for our purposes, (3) platelets are highly perishable. Platelets have a shelf life of only 5 days from the time of donation; and since the first 2 days are taken up by transportation, testing, and processing, the shelf life is really only 3 days once the platelets reach a hospital.

Most hospitals order platelets from blood banks every 1–2 days. If they only ordered every 3 days, the inventory problem would be easy—it would be equivalent to an infinite-horizon newsvendor problem, since the inventory in each replenishment order expires just before the next order is placed. But the fact that the order interval is shorter than the shelf life complicates the analysis.

As a starting point, consider the periodic-review models of Section 4.3. The infinite-horizon model (Section 4.3.4) is not appropriate here because it assumes the demands are stationary. Therefore, the finite-horizon model (Section 4.3.3) will be the basis for our platelet model.

First consider a generic model for a product that has a shelf life of m periods: Items that arrive in period t expire at the end of period $t + m - 1$. (So, for the newsvendor problem, $m = 1$.) All inventory arrives new, and inventory is used according to a first-in, first-out (FIFO) policy, i.e., oldest first. This model was introduced by Nahmias (1975).

We'll consider the lost sales case, which makes the accounting a bit easier, but the model can be modified for backorders if desired. In addition to the usual per-unit ordering, holding, and stockout costs, there is a cost of b per item that expires, called the *outdate cost*. We will use the same sequence of events as in Section 4.3. With the nonstationarity of platelet use in mind, one could also allow the demand distribution to vary over time, with $f_t(\cdot)$ representing the pdf of the demand in period t, though we will assume stationarity to keep things simpler.

Let x_i be the inventory on hand at the beginning of a given period that will expire in exactly i periods (we call these "type-i" units), and let $\mathbf{x} = \{x_1, \ldots, x_{m-1}\}$ be the corresponding vector. Thus, the inventory vector at the start of period t is $\mathbf{x}$, then x_1 units will expire at the end of period t, x_2 will expire at the end of $t + 1$, and so on. To formulate the DP recursion, we will need to know the starting inventory vector for period $t + 1$, given $\mathbf{x}$, the order-up-to quantity y, and the demand d. This vector is given by a (vector-valued) function denoted $\mathbf{s}(\mathbf{x}, y, d)$, which is characterized in the next lemma. That lemma also identifies the number of units that expire in each period, which we need in order to calculate the outdate cost.

Lemma 16.1 *Let $\mathbf{x}$ be the inventory vector at the start of period t. Suppose we order up to y in period t and experience a demand of d. Then, for all $1 \le i \le m$:*

(a) The number of units of type-i inventory remaining at the end of period t is

$$\left[x_i - \left(d - \sum_{j=1}^{i-1} x_j\right)^+\right]^+.$$

If $i = 1$, then these units expire. If $i > 1$, then

$$s_{i-1}(\mathbf{x}, y, d) = \left[x_i - \left(d - \sum_{j=1}^{i-1} x_j \right)^+ \right]^+ .$$

(b) *After using the type-1 through type-i units to satisfy demands in period t (following the FIFO policy), there are*

$$\left(d - \sum_{j=1}^{i} x_j \right)^+$$

units of demand still remaining. If $i = m$, these demands become stockouts.

We interpret x_m (which occurs when $i = m$ in parts (a) and (b)) as the number of units that were ordered in period t, i.e., $x_m = y - \sum_{j=1}^{m-1} x_j$. These "fresh" units will expire at the end of period $t + m - 1$. In part (a), if $i = 1$, then the sum $\sum_{j=1}^{i-1} x_j$ is taken to equal 0.

Proof. By induction on i. First consider the base case, $i = 1$. There are x_1 type-1 units on hand. After using some to satisfy the demand, there will be $(x_1 - d)^+$ left, and they will expire, proving the base case for part (a). Moreover, $(d - x_1)^+$ demands will still remain, to be filled (to the extent possible) by the remaining types of inventory; this proves the base case for part (b).

Now suppose the lemma holds for some i, $1 \le i \le m - 1$; we will show they hold for $i + 1$. By the induction hypothesis for part (b), after the type-1 through type-i units have been used in period t, there are

$$\left(d - \sum_{j=1}^{i} x_j \right)^+$$

demands remaining to be satisfied. Type-$(i + 1)$ units will be used next, after which there will be

$$\left[x_{i+1} - \left(d - \sum_{j=1}^{i} x_j \right)^+ \right]^+$$

type-$(i + 1)$ units still in inventory. At the start of period $t + 1$, these units age by one period, becoming type-i units. That is,

$$s_i(\mathbf{x}, y, d) = \left[x_{i+1} - \left(d - \sum_{j=1}^{i} x_j \right)^+ \right]^+ ,$$

confirming the induction step for part (a). Next, the number of demands remaining after type-1 through type-$(i + 1)$ units have been used is given by

$$\left[\left(d - \sum_{j=1}^{i} x_j \right)^+ - x_{i+1} \right]^+ = \left(d - \sum_{j=1}^{i+1} x_j \right)^+$$

by (C.7). Finally, if $i = m - 1$ (so $i + 1 = m$), then these demands become stockouts because there are no more types of inventory that can be used to meet the demand. This proves the induction step for part (b). ■

Note that if $i = m$, then Lemma 16.1(b) just says that the number of stockouts in period t is $(d - y)^+$, as usual. Similarly, the period-t ending inventory is simply $(y - d)^+$, as usual. In other words, the function $g(y)$ from (4.37) still gives the expected holding and stockout cost. By Lemma 16.1(a), the expected outdate cost is

$$b\mathbb{E}_D[(x_1 - D)^+] = b \int_0^{x_1} (x_1 - d) f(d) dd.$$

Let $\theta_t(\mathbf{x})$ be the optimal total expected cost in periods $t, \ldots, T$ if we start period t with inventory vector $\mathbf{x}$. Then, using the ideas in Section 4.3 and Lemma 16.1, we can express $\theta_t(\mathbf{x})$ recursively as

$$\theta_t(\mathbf{x}) = \min_{y \geq x} \left\{ c(y - x) + g(y) + b \int_0^{x_1} (x_1 - d) f(d) dd + \gamma \mathbb{E}_D \left[\theta_{t+1} \left(\mathbf{s}(\mathbf{x}, y, D) \right) \right] \right\}. \tag{16.23}$$

(As usual, x represents the starting inventory, i.e., $x = \sum_{i=1}^{m-1} x_i$.)

Equation (16.23) seems familiar, with a few new twists, most notably the use of the function $\mathbf{s}(\cdot)$ to relate the ending inventory in period t to the starting inventory in period $t+1$. Moreover, Nahmias (1975) proves that the function inside the $\{\cdot\}$ in (16.23) is convex in y (although, unlike in the model in Section 4.3, $\theta_t(\mathbf{x})$ is not convex in $\mathbf{x}$!). Therefore, the form of the optimal policy can be characterized (see Nahmias (1975)), though its structure can be quite complex.

This should all be good news. Unfortunately, however, things are not so simple, because of the curse of dimensionality: The size of the state space explodes rapidly with m, since the state space has dimension $m-1$. For example, fresh milk has a shelf life at the grocery store of roughly 2 weeks. If the grocery store never orders more than, say, a dozen cartons of milk per day (i.e., $x_i \leq 12$), then the state space has $12^{m-1} = 12^{13} = 106{,}993{,}205{,}379{,}072$ elements! And for each possible state, we need to solve the minimization problem in (16.23), and we need to do this for every period in the time horizon.

For this reason, the model above, introduced in 1975, is not often implemented in practice. Some simpler settings have been considered: For example, Nahmias and Pierskalla (1973) consider $m = 2$, which means the state variable has only one dimension. Heuristics have also been developed for the m-period problem (Nandakumar and Morton 1993). And of course, if $m = 1$, then we simply have a newsvendor problem.

All of which brings us back to blood platelets. Since platelets have a shelf life of 3 days, we have a reasonable value of m, making Nahmias's (1975) model practical. Zhou et al. (2011) study the inventory of platelets specifically, assuming that the hospital places "regular" replenishment orders for platelets every 2 days but can also place "emergency" orders on the off-days. They use this model to provide guidance to hospitals about whether they should order every day or every 2 days, based on the demand and cost parameters. See also Prastacos (1984) for a review of OR models for blood inventory management.

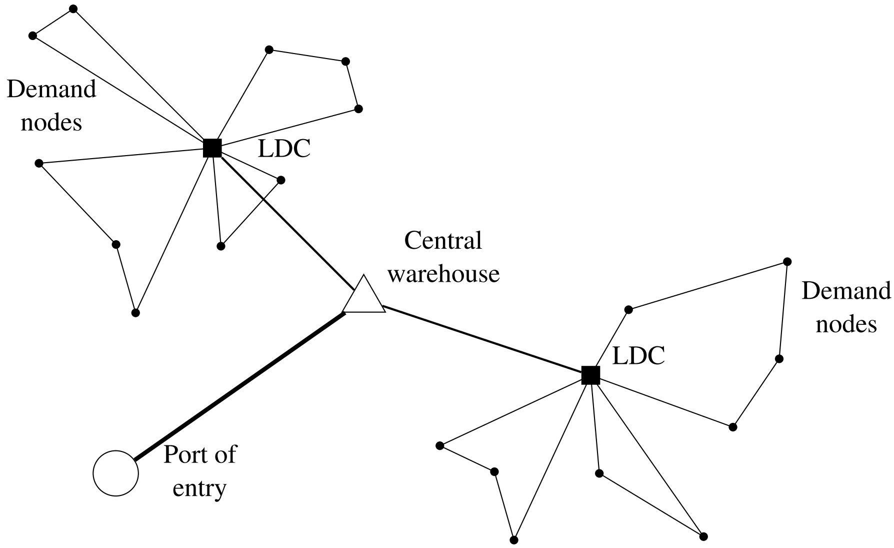


Figure 16.3 Typical disaster-relief supply chain.

16.4 PUBLIC SECTOR OPERATIONS

Disaster relief, homeland security, public services planning, and other types of public sector operations have become important applications of OR and of supply chain theory in particular. These problems involve complex networks, allocation of scarce resources, movements of physical goods, and inherent uncertainty—all characteristics that the tools of supply chain theory are well suited to address. In this section, we discuss three topics that make use of these tools. For further reading, see the reviews by Wright et al. (2006), Altay and Green (2006), Johnson and Smilowitz (2007), Caunhye et al. (2012), Çelik et al. (2012), and McLay (2015).

16.4.1 Disaster Relief Routing

After a natural or manmade disaster, it is critical to distribute emergency supplies to affected people quickly and efficiently. The supply chain that enables this distribution is usually established, at least in part, in real time immediately following the disaster. Further complicating the planning process are logistical complications such as limited resources, damaged roads, and multiple agencies coordinating (sometimes poorly) to provide relief.

Figure 16.3 depicts a typical disaster-relief supply chain. Supplies arrive in the affected country from around the world via a port of entry, from which they are shipped to a central warehouse. From there, supplies are shipped to local distribution centers (LDCs), which are often established in makeshift locations such as schools or warehouses, or even temporary facilities such as tents. Deliveries are made from LDCs to the recipients of emergency aid via less-than-truckload (LTL) shipments consisting of multiple stops per route. This last portion of the supply chain is referred to as *last-mile* distribution.

Balcik et al. (2008) introduce a model for last-mile disaster relief distribution. Their model considers both resource allocation (how many supplies to deliver to each demand node, and when) and vehicle routing (how to route the delivery vehicles to the demand nodes). Because of the resource allocation component, their model resembles the inventory–routing problem (IRP) from Section 12.4 more closely than the vehicle routing problem (VRP) from Chapter 11. However, unlike the IRP formulation in Section 12.4, which makes yes/no decisions for individual edges of the network in order to construct the routes, the model by Balcik et al. (2008) assumes the potential routes have already been identified, and it selects routes from among them. This is especially practical in developing countries in which the road network is not highly connected, so that the total number of potential routes is not large (VonAchen et al. 2016).

Assume that a set of demand nodes has already been assigned to each LDC, so that we can solve the problem for each LDC separately. The model considers a finite planning horizon, during which we may make one or more deliveries to each demand node. Emergency supplies are often categorized as either Type 1 items such as tarps and blankets, which are delivered once at the beginning of the horizon, or Type 2 items such as food and water, which are consumed and therefore must be delivered throughout the horizon. Balcik et al. (2008) consider both types of supplies in their model, but for the sake of simplicity, we will consider only Type 2 items. We will consider a single product, which is used to model all Type 2 items in aggregate.

In disaster relief supply chains, there is an insistence on equity in resource allocation and a priority placed on serving vulnerable groups. In this model, unmet demands are assumed to be lost, rather than backordered, but the stockout penalty is based on the *maximum weighted unmet demand fraction* among all demand nodes. Thus, it is considered preferable for every node to have 10% of its demands unmet than for one node to have 20% of its demands unmet and every other node to have all of its demands satisfied. This type of objective would be unusual in commercial supply chains, which are typically driven by cost or revenue, not equity. Moreover, the model allows some nodes to be weighted more than others when calculating the maximum unmet demand fraction, so that priority may be placed on, for example, vulnerable populations. (See Huang et al. (2012) for further discussion of equity in disaster relief routing.)

A second characteristic of disaster relief distribution problems that differs from commercial supply chains is the inherent uncertainty in demands, supplies, and even the duration of the planning horizon itself. Rather than modeling this uncertainty explicitly using scenarios (as in Section 8.6), Balcik et al. (2008) propose a rolling-horizon approach in which we solve the model for a finite horizon, implement the solution for the first period, then update the estimates of the uncertain parameters, shift the horizon by one period, and solve the model again.

Another difference between relief and commercial supply chains is that in a disaster, it is not always possible to use larger, more cost-effective trucks to deliver all supplies, since those trucks may not be able to travel on certain roads that have been damaged by the disaster. To model this, we assume that each vehicle in the fleet has its own capacity, speed, and ability to travel on each link.

The model assumes that all possible clusters of demand nodes have been enumerated, and for each cluster, a traveling salesman problem (TSP) has been solved to determine the shortest route through those nodes beginning and ending at the LDC. (Of course, this is only practical if the number of demand nodes is reasonably small; otherwise, heuristics can be used to identify good clusters, and good routes for each.) The resulting list of routes is an input to the model. We assume that each vehicle can complete multiple routes in one period, and each demand node can be visited multiple times in one period. Each route is included in the list multiple times, once for each time it can be traveled in one period. For example, if one period is 24 hours long, then a 15-hour route is included once, but a 10-hour route is included twice.

The costs of the model include transportation costs from the LDC to demand nodes, as well as a stockout cost for unmet demand (which is assumed to be lost rather than backordered). Excess inventory can be held at a demand point from one period to another, but the holding cost is assumed to be negligible compared to the other costs and is therefore ignored.

We use the following notation:

Sets

I = set of demand nodes
K = set of available vehicles
R = set of routes
$I(r)$ = set of demand nodes included on route $r \in R$
T = set of time periods

Parameters

Costs

c_{rk} = cost of route $r \in R$ for vehicle $k \in K$
p_{it} = stockout weight at node $i \in I$ in period $t \in T$

Other

h_{it} = demand of node $i \in I$ in period $t \in T$
a_t = supply delivered to LDC at the beginning of period $t \in T$
D_k = capacity of vehicle $k \in K$
τ_{rk} = fraction of one period that route $r \in R$ requires when using vehicle $k \in K$

Decision Variables

x_{rtk} = 1 if route $r \in R$ is traveled by vehicle $k \in K$ in period $t \in T$, 0 otherwise
y_{irtk} = amount of supplies delivered to node $i \in I$ by vehicle $k \in K$ via route $r \in R$ in period $t \in T$
w_t = stockout penalty in period $t \in T$
s_{it} = fraction of demand at node $i \in I$ that is unsatisfied in period $t \in T$
I_{it} = on-hand inventory at location $i \in I$ at the beginning of period $t \in T$

The relief routing problem can then be formulated as follows:

$$\text{minimize} \quad \sum_{t\in T}\sum_{r\in R}\sum_{k\in K} c_{rk}x_{rtk} + \sum_{t\in T} w_t \tag{16.24}$$

$$\text{subject to} \quad \sum_{r\in R}\sum_{i\in N(r)}\sum_{s=1}^{t}\sum_{k\in K} y_{irsk} \le \sum_{s=1}^{t} a_s \qquad \forall t \in T \tag{16.25}$$

$$w_t \ge p_{it}s_{it} \qquad \forall i \in I, \forall t \in T \tag{16.26}$$

$$\frac{1}{h_{it}}\left(h_{it} - \sum_{r:i\in N(r)}\sum_{k\in K} y_{irtk} - I_{it} + I_{i,t+1}\right) = s_{it} \qquad \forall i \in I, \forall t \in T \tag{16.27}$$

$$\sum_{i \in N(r)} y_{irtk} \leq D_k x_{rtk} \quad \forall r \in R, \forall t \in T, \forall k \in K \tag{16.28}$$

$$\sum_{r \in R} \tau_{rk} x_{rtk} \leq 1 \quad \forall t \in T, \forall k \in K \tag{16.29}$$

$$I_{it}, s_{it} \geq 0 \quad \forall i \in I, \forall t \in T \tag{16.30}$$

$$y_{irtk} \geq 0 \quad \forall i \in I, \forall r \in R, \forall t \in T, \forall k \in K \tag{16.31}$$

$$x_{rtk} \in \{0, 1\} \quad \forall r \in R, \forall t \in T, \forall k \in K \tag{16.32}$$

The objective function (16.24) calculates the total transportation and stockout cost. Constraints (16.25) enforce the available supply at the LDC in every time period: The total amount shipped to every demand node by every vehicle on every route in every period up through time t cannot exceed the total supply delivered to the LDC up through time t. Constraints (16.26) set the stockout penalty in period t, w_t, equal to the maximum of the weighted stockout fraction over all nodes in period t. Note that s_{it} is a fraction, but p_{it} can be scaled up or down to convert the stockouts to an equivalent value in currency so that it can be added to the objective function.

The quantity inside the parentheses in constraints (16.27) is equal to the unmet demand at node i in period t: It equals the demand minus the starting inventory and arriving shipments, adding back any units that are reserved for period $t + 1$. Constraints (16.27) therefore set s_{it} equal to the fraction of node i's demand that is unmet in period t.

Constraints (16.28) enforce the vehicle capacity, while constraints (16.29) ensure that the routes traveled by vehicle k in a given period do not exceed the period length. Constraints (16.30)–(16.32) are nonnegativity and integrality constraints.

Balcik et al. (2008) solve the model using an off-the-shelf MIP solver. Even for small instances, though, the model takes a few hours to solve.

16.4.2 Passenger Screening

Consider an aviation security agency, such as the United States Transportation Security Administration (TSA), that wishes to determine which security *classes* to assign passengers to. A class consists of the equipment (such as metal detectors) and procedures (such as manual baggage inspections) that will be used to screen passengers assigned to that class. Each class that is selected for use involves a fixed cost to purchase the equipment, train the security agents, and so on, as well as a per-unit cost for each passenger assigned to that class.

For example, one class might consist of screening the passenger using a metal detector, screening the passenger's carry-on baggage using an x-ray machine, and screening his or her checked bags using an explosive detection system (EDS). Another class might consist of the same three screenings, plus a hand-wand inspection for the passenger and a detailed hand search of the carry-on baggage. A third class might consist of the same as the first class plus an open-bag trace (in which the bag is tested for traces of explosive-related chemicals using a swab) for the carry-on baggage and an open-bag trace plus a detailed hand search for the checked baggage. The agency may choose to employ some or all of these classes but can only assign passengers to a class that has been chosen. There is a fixed cost for each class chosen, which implicitly assumes that each class has its own dedicated equipment; if two classes each use an EDS, for example, they must each have their own dedicated EDS equipment.

Assume that the agency knows the number of passengers traveling through a given airport at a typical peak hour, as well as the assessed threat level of each passenger. (*How* these threat levels are assessed is a highly sensitive subject, and rife with controversy, but is outside the scope of our discussion.) The agency's goal is to decide which security classes to use, and which passengers to assign to each class, in order to maximize the security provided while respecting a budget constraint.

In this section, we will formulate a discrete optimization model to solve this problem. The model was introduced by McLay et al. (2006). This and other related papers provided the analysis that laid the groundwork for the TSA's PreCheck program (McLay 2015).

Let I be the set of passengers in a typical hour and let J be the set of potential security classes. Let a_i be the assessed threat level of passenger i, and let s_j be the security level achieved by class j. If passenger i is assigned to class j, the resulting security level is calculated as $a_i s_j$, and our goal is to maximize the sum of this value over all passengers. Therefore, the objective encourages passengers with high assessed threat levels to be assigned to classes with high security levels.

The fixed cost for security class j is denoted by f_j, and the per-passenger cost is c_j. The fixed cost includes the initial investment cost, divided by the total number of hours in the equipment's expected useful life, plus the hourly cost to operate the equipment. The agency has a fixed budget of b available to spend on these costs per hour.

There are two sets of decision variables in the model:

x_j = 1 if class j is chosen, 0 otherwise
y_{ij} = 1 if passenger i is assigned to class j, 0 otherwise

$$\text{maximize} \quad \sum_{i\in I}\sum_{j\in J} s_j a_i y_{ij} \tag{16.33}$$

$$\text{subject to} \quad \sum_{j\in J} y_{ij} = 1 \qquad \forall i \in I \tag{16.34}$$

$$y_{ij} \le x_j \qquad \forall i \in I, \forall j \in J \tag{16.35}$$

$$\sum_{j\in J} f_j x_j + \sum_{i\in I}\sum_{j\in J} c_j y_{ij} \le b \tag{16.36}$$

$$x_j, y_{ij} \in \{0,1\} \qquad \forall i \in I, \forall j \in J \tag{16.37}$$

The objective function (16.33) maximizes the total security level. Constraints (16.34) require every passenger to be assigned to a class, and constraints (16.35) prevent a passenger from being assigned to a class that has not been chosen. Constraint (16.36) is the budget constraint. Constraints (16.37) are integrality constraints.

This model is very similar to the uncapacitated fixed-charge location problem (UFLP) from Section 8.2. The main differences are that the total cost is bounded in the constraints, rather than minimized in the objective, and that the objective instead maximizes the security. The per-unit cost is also slightly different, since it depends only on the screening class and not on the passenger; that is, it is written c_j instead of c_{ij}. Moreover, there is no h_i term, as in the UFLP, because the passengers are assumed to be enumerated in the set I. We could instead define I as the set of passenger *types*, in which case h_i might represent the number of passengers of each type, and we would multiply the objective function (16.33) and the per-unit cost in (16.36) by h_i.

On the other hand, the presence of the budget constraint (16.36) makes this problem harder to solve than the UFLP. McLay et al. (2006) propose a greedy heuristic to solve the problem, as well as a dynamic programming (DP) algorithm that can be used when the number of classes is relatively small.

The model above is static, in the sense that it considers a single time period and assumes that passenger threat levels are known. In contrast, dynamic screening models assume a sequential process in which a passenger's threat level is not determined until she or he reaches a security checkpoint. McLay et al. (2009, 2010) and Nikolaev et al. (2007) formulate such models as MDPs.

16.4.3 Public Housing Location

Public housing authorities (PHAs) provide affordable housing for low-income residents of a city or other area. For many years, the predominant strategy used by PHAs in large cities in the United States was to build large, high-rise housing developments. Such developments were cheaper to build and maintain due to economies of scale, but they gained a reputation for fostering crime and gang violence, as well as leading to increased racial and socio-economic segregation. More recently, U.S. PHAs have favored lower-density housing projects that are spread more widely throughout the city.

Although PHAs have not typically used OR methods for decision-making, the problem of deciding where to locate public housing throughout a city is a natural facility location problem. It is made more complicated by the fact that the decision involves multiple stakeholders with competing objectives: The PHA might want fewer, larger projects; the residents want locations that offer good schools and other benefits; society as a whole wants integrated, diverse communities; while at the same time, many neighbors of potential housing projects do not want the project to be built at all. The political forces, social controversy, and moral questions surrounding public housing are very difficult to quantify, but once an attempt has been made to do so, OR can offer tools to help navigate these issues in choosing project locations.

Johnson (2006) introduces a multiobjective facility location model to choose where to locate public subsidized housing projects and how many housing units to build at each location. The first objective is the *economic efficiency*, defined as the difference between the costs of constructing the facilities and the economic benefits accrued by the residents, the neighboring community, and society as a whole. (The costs may be straightforward to measure, but the benefits, of course, are not. They also tend to be nonlinear, but we will treat them as linear.) The second objective is the *perceived equity* among the various stakeholders—another metric that is difficult to quantify and that is typically nonlinear. Johnson (2006) proposes, as a proxy, a dispersion objective, i.e., maximizing the distances among the projects located. (See Section 8.5.1.)

Let J be the set of potential locations for the housing projects, and let d_{jk} be the distance between locations j and k in J. Let f_j be the fixed cost to build a project at location j, and let b_j be the net economic benefit for each unit built there. (That is, b_j is the benefit accruing from one unit of housing built in a project at location j minus the cost of building the unit.) Assume that the PHA wishes to build a total of H units, and that at most H_j units can be built at location j.

The model uses decision variables x_j, which equals 1 if we build a project at location $j \in J$, 0 otherwise; and y_j, which is the number of units to build at location j. It also uses a set of auxiliary variables δ_{jk}, which equals d_{jk} if we build projects at both locations j and k, and equals a large number otherwise.

Johnson (2006) formulates the public housing location problem as follows:

$$\text{minimize} \quad \sum_{j \in J} (f_j x_j - b_j y_j) \tag{16.38}$$

$$\text{maximize} \quad \min_{j,k \in J} \{\delta_{jk}\} \tag{16.39}$$

$$\text{subject to} \quad \delta_{jk} = d_{jk} + M(1 - x_j) + M(1 - x_k) \qquad \forall j \in J, \forall k \in J \tag{16.40}$$

$$\sum_{j \in J} y_j = H \tag{16.41}$$

$$y_j \le H_j x_j \qquad \forall j \in J \tag{16.42}$$

$$x_j \in \{0, 1\} \qquad \forall j \in J \tag{16.43}$$

$$y_j \ge 0 \qquad \forall j \in J \tag{16.44}$$

The model contains two objective functions. The first (16.38) calculates the economic efficiency of the projects and units that are built. The second (16.39) maximizes the minimum distance between any two housing projects. By constraints (16.40), $\delta_{jk} = d_{jk}$ if $x_j = x_k = 1$, and δ_{jk} equals a large number otherwise. (M is a large constant.) Thus, if x_j or x_k equals 0, δ_{jk} has no effect in (16.39). Constraint (16.41) ensures that exactly H units will be built. Constraints (16.42) enforce the capacity-type restriction that no more than H_j units can be built at location j if a project is located there ($x_j = 1$) and prevents any units from being built there if $x_j = 0$. Constraints (16.43) and (16.44) are integrality and nonnegativity constraints.

To transform this model into a single-objective model that can be solved using an off-the-shelf MIP solver, Johnson (2006) proposes using the *weighting method* (Cohon 1978), in which the two objectives are combined into a single objective function by taking a linear combination of them. (We discussed a similar approach in Section 9.6.5.) By varying the weight systematically, we obtain a trade-off curve.

This model is NP-hard and is much more difficult to solve, computationally, than many other NP-hard facility location problems such as the UFLP (Section 8.2). Johnson (2006) reports CPLEX computation times of 4 hours or more for a case study with $|J| = 50$. That case study uses economic data for Cook County, Illinois to explore the model and its outcomes. The resulting trade-off curve, unfortunately, does not contain as many points as the curve in Figure 9.11, meaning that decision-makers have less flexibility in navigating the competing objectives. It also does not have as sharp an "elbow," i.e., it requires more of a sacrifice in one objective in order to improve the other. Nevertheless, the model is still valuable as a planning tool and can allow PHAs to make data-driven decisions in a flexible and equitable framework.

CASE STUDY 16.1 Optimization of the Natural Gas Supply Chain in China

Natural gas has become an important energy resource and economic driver, especially in China. The annual growth rate of China's natural gas industry is roughly 14%. Rapid increases in the rates of production and consumption have made supply chain management a significant issue in the natural gas industry. The supply chain of natural gas includes many of the functions discussed in this book, such as production, storage, network design, transportation, and sales. In China, decision makers make centralized decisions about all of these elements of the supply chain.

The management of China's natural gas supply chain is complex for two main reasons. First, there are thousands of demand nodes in the network, and the network itself is interconnected and contains many cycles. Second, the physical laws governing the relationship between flow and pressure in a pipeline are highly nonlinear and nonconvex. (This echoes the laws governing power flows in electricity systems from Section 16.2.) This makes it difficult to answer even simple questions about the system, such as whether the network has sufficient capacity to meet the nation's increasing demand.

The China National Petroleum Corporation (CNPC), China's largest oil and natural gas producer and supplier, controls 75% of the country's natural gas resources and pipeline network. Yet planners at the company were conducting most of their planning manually using spreadsheets, a process that tended to result in large optimization errors, increased costs, and wasted resources. Therefore, CNPC partnered with researchers at the University of California, Berkeley and at Tsinghua University to develop and implement software to optimize the country's pipeline. Their work was a finalist for the INFORMS Edelman Award in 2018 (INFORMS 2018). We summarize the model here; for more details, see Han et al. (2019). We focus on the optimization of production, procurement, transmission, and sales, given a fixed pipeline network topology.

The optimization of natural gas transmission is similar to conventional transportation or network flow problems—such as the problem that arises in arc design problems (Section 8.7.2) after the yes/no decisions on the arcs have been made—because both problems involve flow-balance constraints, such as (8.144). However, natural gas pipeline flows are much more complex than in classical network flow problems because we must explicitly model the pipeline's flow rates as a function of the pressure and temperature in the network, while satisfying other network flow constraints and pipeline constraints.

We consider the optimization problem from a central planner's perspective. The planner decides how much gas to purchase or produce at each location. Our goal is to maximize the total profit, which can be calculated as the sales revenue minus the production cost:

$$\text{maximize} \quad \sum_{i \in N_d} c_i^d(-s_i) - \sum_{i \in N_s} c_i^s s_i,$$

where we use the following notation:

Sets

N	= set of nodes in the network
N_s, N_d	= sets of supply and demand nodes (respectively); $N_s, N_d \subseteq N$
A	= set of arcs in the network
A_p, A_c, A_r	= sets of pipelines, compressors, and regulator valves (resp.); $A_p, A_c, A_r \subseteq A$

Parameters

c_i^s	= cost per unit to produce or purchase natural gas at node i
c_i^d	= revenue per unit to sell natural gas at node i
$\underline{s}_i$, $\overline{s}_i$	= lower and upper bounds (resp.) on supply or demand at node i
$\underline{p}_i$, $\overline{p}_i$	= lower and upper bounds (resp.) on pressure at node i
α_{ij}	= parameter related to change of elevation on pipeline $(i,j) \in A_p$
β_{ij}	= parameter related to flow resistance on pipeline $(i,j) \in A_p$

Decision Variables

s_i	= net supply: gas produced/purchased (> 0) or sold (< 0) at node i
p_i	= natural gas pressure at node i
π_i	= square of pressure at node i
f_{ij}	= natural gas volumetric flow per unit of time on arc $(i,j) \in A$

The constraints consist of network flow constraints and pipeline constraints. Network flow constraints ensure that the gas flow balances at each node, that the supply and demand fall into the right ranges, and that the gas flow and pressure are controlled within safety ranges:

$$\sum_{j:(i,j)\in A} f_{ij} - \sum_{j:(j,i)\in A} f_{ji} = s_i \quad \forall i \in N \tag{16.45}$$

$$\underline{s}_i \le s_i \le \overline{s}_i \quad \forall i \in N \tag{16.46}$$

$$\underline{p}_i^2 \le \pi_i \le \overline{p}_i^2 \quad \forall i \in N \tag{16.47}$$

Constraints (16.45) are the flow balance constraints and are similar to (8.144) in the arc design model and (16.15) in the electricity network design model. Similar to the variable p_{ij} in that model, the variable f_{ij} here can be positive or negative, and the sign indicates the direction of flow: If $f_{ij} > 0$, then gas is flowing from i to j. Using π_i, rather than p_i^2, in (16.47) and other constraints below helps to avoid squared terms where possible.

In the network, special types of arcs are used to change the pressure of natural gas: Compressors add pressure, while regulator valves reduce it. We simplify the relationship between the pressure at the two end nodes of a compressor or regulator valve by formulating them as linear constraints:

$$\pi_i \le \pi_j \quad \forall (i,j) \in A_c \tag{16.48}$$

$$\pi_i \ge \pi_j \quad \forall (i,j) \in A_r. \tag{16.49}$$

The flow rate on a pipeline can be expressed as a function of the pressure and temperature of the natural gas at the two ends of the pipeline. Engineers describe the relationship among flow rates, pressure, and temperature of natural gas along a pipeline (i,j) as follows:

$$\frac{\tilde{\lambda}\tilde{Z}\tilde{\Delta}\tilde{T}\tilde{L}\left[1 + \frac{\tilde{\alpha}}{2\tilde{L}}\sum_{j=1}^{\tilde{n}}\left(\tilde{h}_j + \tilde{h}_{j-1}\right)\tilde{L}_j\right]}{1051^2\tilde{d}^5} f_{ij}^2 = p_i^2 - \left(1 + \tilde{\alpha}\Delta\tilde{h}\right)p_j^2 \tag{16.50}$$

All of the parameters with tildes in (16.50) have physical meanings. For example, $\tilde{h}$ represents the height of a node; $\Delta\tilde{h}$, $\tilde{L}$, and $\tilde{d}$ represent the elevation change, length, and diameter of the pipeline, respectively; and $\tilde{\lambda}$, $\tilde{Z}$, and $\tilde{\Delta}$ are related to the state of the natural gas. We simplify (16.50) by replacing the fraction on the left-hand side with β_{ij} and the term inside the parentheses on the right-hand side with α_{ij}. We also replace the square of the pressure, p_i^2, with π_i, to get the following constraint, which is included in the optimization model:

$$\beta_{ij} f_{ij}^2 = \pi_i - \alpha_{ij}\pi_j \quad \forall (i,j) \in A_p. \tag{16.51}$$

In practical settings, β_{ij} changes with the pressure and temperature at the two end nodes and is iteratively updated when solving the problem.

De Wolf and Smeers (2000) and Babonneau et al. (2012) propose a piecewise-linear approximation method that can be modified and used to deal with the nonlinearity of (16.51); see Han et al. (2019). The basic idea is to find several initial breakpoints using a special initialization model and then piecewise-linearize the f_{ij}^2 terms. The initialization problem is the dual of the original model with only the constraints on the flow rates of natural gas, which is called the energy minimization problem. The set of breakpoints is updated iteratively to narrow the gap between the resulting solutions and the optimal ones.

Han et al. (2019) also introduce a three-stage convex relaxation method to deal with the nonconvexity of (16.51). In the first stage, (16.51) is relaxed to an inequality, and the relaxed (convex) model is solved. If the results are infeasible, then an energy minimization model is solved to obtain feasible flow rates. In the third stage, the pressure and temperature of natural gas are determined given the flow rates from the second stage.

CNPC implemented the model developed by the Berkeley and Tsinghua researchers at the end of 2014. Since then, CNPC has realized roughly \$330 million in direct savings. In addition, the increased efficiency of the pipeline system has enabled CNPC to delay further pipeline expansions, saving billions of additional dollars.

PROBLEMS

16.1 (Single-Period Behind-the-Meter Problem) Show that the behind-the-meter energy storage problem from Section 16.2.1.1 is equivalent to a newsvendor problem (possibly plus a constant) if $T = 1$.

16.2 (Single-Period Wind Farm Problem) Show that the expected cost function (16.4) is equivalent to a newsvendor problem (possibly plus a constant) by giving values for h and p, in terms of the notation in Section 16.2.1.2, that make the two cost functions equal (except for an additive constant). What is the optimal bid quantity?

16.3 (Selling on the Real-Time Market) Modify the single-period wind farm problem from Section 16.2.1.2 to allow the wind farm operator to sell power to the real-time market if the observed wind power is greater than its bid on the day-ahead market. Is the problem still equivalent to a newsvendor problem? If so, what is the optimal solution?

16.4 (Multi-Period Selling on the Real-Time Market) Modify the recursion (16.7) to allow the wind farm operator to sell power to the real-time market in each period if it wishes.

16.5 (Vaccine Supplier's Expected Cost Function) Prove that (16.21) equals (16.20).

16.6 (System-Optimal Solution for Vaccine Problem) In the vaccine model in Section 16.3.1, suppose that the function $T(f)$ is approximated using the following piecewise-linear function:

$$T(f) = \begin{cases} M - N\psi f, & \text{if } 0 \le f \le f^0 \\ 0, & \text{if } f^0 \le f \le 1, \end{cases} \tag{16.52}$$

where M and ψ are nonnegative constants. (This is the function plotted in Figure 16.2.) Assume that $\psi b - c_g\delta > 0$ (so that the expected benefit from vaccination exceeds the cost).

a) Prove that the system objective function $G(Q, f)$ (i.e., $G(Q, f) = g_s(Q, f) + g_g(Q, f)$) is given by

$$G(Q, f) = c_s Q + \mathbb{E}\left[bT\left(\frac{V}{N\delta}\right) + c_g V\right],$$

where

$$V = \min\{QZ, fN\delta, f^0 N\delta\}.$$

Now, let (Q^*, f^*) be the solution that minimizes $G(Q, f)$.

b) Prove that all values of f^* in $[f^0, 1]$ are optimal for the system.

c) Prove that Q^* satisfies

$$\int_0^{f^0 N\delta/Q^*} z f_Z(z) dz = \frac{c_s}{\psi b/\delta - c_g}.$$

16.7 (Shared Security Equipment) The passenger screening model in Section 16.4.2 assumes that each security class uses its own dedicated equipment. Relax this assumption by assuming that there is a set of equipment types and that each security class uses a predefined subset of those types. The fixed and per-unit costs apply to the equipment, rather than to the classes. Formulate a model that decides what equipment to purchase and which classes to assign each passenger to, in order to maximize the total security, while respecting a budget constraint. You will have to introduce new notation; define it clearly. Explain your objective function and constraints in words.

16.8 (Integral Number of Housing Units) In the public housing location problem in Section 16.4.3, we defined y_j, the number of housing units to build at location j, as a continuous decision variable. Does there always exist an optimal solution to this model in which y_j will be a nonnegative integer? Justify your answer.

APPENDIX A

MULTIPLE-CHAPTER PROBLEMS

PROBLEMS

A.1 (Worst-Case Bound for Deterministic Newsvendor Approximation) As noted in Section 5.3.2, the papers by Zheng (1992) and Axsäter (1996) suggest bounds on the error that results from approximating a stochastic inventory model (the model of Section 5.1, for which an (r, Q) policy is optimal) by a deterministic one. Suppose we do the same thing for the newsvendor model, setting S equal to the optimal solution to the deterministic problem, i.e., $S = \mu$. Assume the demand is distributed $N(\mu, \sigma^2)$.

a) Prove that

$$\rho \equiv \frac{g(\mu)}{g(S^*)} \approx \frac{0.3989}{\phi(z_\alpha)},$$

where $g(S)$ is the expected newsvendor cost if S is the order-up-to level and $\alpha = p/(h + p)$.

b) Is it possible to identify a fixed worst-case bound $\bar{\rho}$ that holds for any values of the parameters h, p, μ, and σ? Explain your answer.

A.2 (Optimizing Compost Inventory) The compost facilities in Greentown move finished compost (organic matter that has completed the composting process) from the *processing area* to a large pile in the *pick-up area* for residents of the town to pick up for use in their gardens, free of charge.

Companion website: www.wiley.com/go/Snyder/SupplyChainTheory

Table A.1 Demand for finished compost for Problem A.2(b).

d	Probability $f(d)$	Cumulative Probability $F(d)$
0	0.12	0.12
1	0.05	0.17
2	0.07	0.24
3	0.11	0.35
4	0.24	0.59
5	0.17	0.76
6	0.11	0.87
7	0.07	0.94
8	0.04	0.98
9	0.02	1.00

a) At the compost facility in the Appleville neighborhood of Greentown, residents pick up compost at a steady rate of 700 cubic yards per week. The facility has committed to keeping the pick-up area stocked, i.e., never having residents arrive to find no compost there.

Finished compost is moved from the processing area to the pick-up area by truck, and each time the facility wishes to do this, it must hire a truck driver for a day. This costs \$320, and once the driver is hired, the facility can move as much finished compost as it wishes. You can assume it takes negligible time to move the compost. Finished compost that remains in the pick-up area must be tended by the staff (e.g., cleaning up spills), at a cost of \$0.05 per cubic yard per day.

Using the EOQ model, calculate the optimal order quantity, Q^*, and the optimal average cost per week, $g(Q^*)$.

b) At the compost facility in the Beantown neighborhood of Greentown, the daily demand for compost is stochastic (unlike at the Appleville facility), with probabilities given in Table A.1. At this facility, the staff moves finished compost from the processing site to the pick-up site every morning, before the facility opens to customers. The truck drivers are staff members and so no additional labor charge is incurred for these activities.

If the pick-up area runs out of compost, for the remainder of the day the staff must deliver compost directly to customers' vehicles, at a cost of \$0.25 per cubic yard. The staff do not tend to the pick-up area during the day (as they do at the Appleville facility), but if there is any finished compost in the pick-up area when the facility closes at the end of the day, they must move it back to the processing area, at a cost of \$0.10 per cubic yard.

What is the optimal number of cubic yards of finished compost to move to the pick-up area each morning?

A.3 (Inventory Optimization for Deterministic Bass Demands) A new model of wireless router will be introduced shortly. An electronics retail chain expects the aggregate demand for the router at its stores to follow a discrete-time Bass diffusion process (2.50) with parameters $m = 100,000$, $p = 0.01$, and $q = 0.3$. Each time period represents one week. The retail chain holds inventory of the routers at its central warehouse and has an opportunity to place a replenishment order once per week. Each order incurs a fixed cost of \$800, and each router held in inventory incurs a holding cost of \$0.04 per week. The planning horizon is 52 weeks; any demand after the horizon ends can be ignored. In which weeks should the retailer place orders for routers, and what are the optimal order quantities? What is the total cost?

A.4 (Inventory Optimization for Stochastic Bass Demands) Suppose that the demand for wireless routers in Problem A.3 is now stochastic. The demand in period t is normally distributed with a mean of d_t and a standard deviation of $\sqrt{d_t}$, where d_t is given by the discrete-time Bass model (2.50). Unmet demands are backordered, incurring a stockout cost of \$1.25 per router per week. Assume $c = 0$ and $\gamma = 1$. The other cost parameters, and the Bass parameters, are as given in Problem A.3. Although we assumed in Section 4.5.2.2 that the demand process is stationary, an (s, S) policy is still optimal for this problem with nonstationary demands. Determine the optimal parameters, s_t^* and S_t^*, for every period, and report the values for periods $t = 1, 12, 22, 32, 42$, and 52.

A.5 (Subscription-Selling Newsvendor) Suppose that, rather than selling individual newspapers, the newsvendor sells *subscriptions* to the newspaper. The subscription is a little unusual and works as follows: There are N customers, and the newsvendor must decide which customers to select. Customers typically request multiple newspapers each day, and customer i's daily demand is $D_i \sim N(\mu_i, \sigma_i^2)$. Demands are statistically independent from one customer to another. If customer i is selected, the newsvendor earns a subscription revenue of r_i per day. This revenue is independent of the actual demand.

Just like in the classical newsvendor problem, our newsvendor must decide at the beginning of the day how many newspapers to stock. During the day, random demands are observed from each of the newsvendor's selected customers. At the end of the day, the newsvendor incurs a holding cost of h per unsold newspaper and a stockout cost of p per unmet demand.

Note that the newsvendor must choose his customers, and his order quantity, before demands are observed.

a) Formulate a mathematical programming model to choose customers in order to maximize the expected profit (revenue minus costs) per day. If you introduce any additional notation, define it clearly.

Hint: Your model should not need a decision variable that represents the order quantity.

b) Formulate a polynomial-time algorithm that solves this problem exactly (i.e., not a heuristic). Describe your algorithm step-by-step and explain clearly why it produces the optimal solution every time.

c) What is the complexity of your algorithm (e.g., $O(N^2)$)?

d) Suppose $h = 1$ and $p = 18$. The table below lists parameters for four customers. Using your algorithm, determine the optimal set of customers to serve. Report the optimal set of customers and the resulting expected profit.

A.6 (EOQ with Market Selection) Consider a single production stage that manufactures a single item. (We can equivalently view this "production stage" as a retail ordering process that plans orders for a single item.) Let $I = \{1, \ldots, n\}$ denote a set of potential markets, indexed by i. Producing the product results in a setup cost K and a variable cost c per item

Table A.2 Customer parameters for Problem A.5(d).

i	r_i	μ_i	σ_i
1	10	40	8
2	8	20	3
3	3	25	9
4	11	16	3

procured. Inventory costs are assessed at a rate of h dollars per unit per year. Market i has a constant and deterministic annual demand rate, λ_i. We let r_i denote the per-unit revenue from market i less any variable production and (possibly market-specific) delivery costs. Unlike the standard EOQ model, the producer can choose whether or not to satisfy each market's demand. If the producer chooses to supply a certain market, then it must satisfy all of the demand for that market. Rather than minimizing the average annual cost, as in the EOQ model, we maximize the average annual net contribution to profit.

a) Write an expression for the average annual net contribution to profit and discuss how to solve the corresponding optimization problem. Define any new notation that you introduce.

b) Suppose that we begin with an n-market problem for which an optimal solution selects markets $1, \ldots, k$, where $k < n$, and then consider the same problem with a single additional new market, $n + 1$. Prove the following proposition:

Proposition A.1 *An optimal solution exists for the new $(n+1)$-market problem that selects at least markets $1, \ldots, k$. If this new solution does not select market $n+1$, then the optimal solution is the same for both the n-market and $(n + 1)$-market problems.*

A.7 (EOQ with Random Half Orders, Take 2) Solve Problem 3.25, reinterpreting it as a yield uncertainty problem and using the results of Section 9.3. (*Hint*: Problem 9.15 justifies the use of the yield uncertainty results even for discrete yield distributions.)

A.8 (Finite-Horizon Transshipments) Consider a finite-horizon version of the (infinite-horizon) transshipment problem discussed in Section 7.4. In the finite-horizon version of the problem, the long-run expectations we calculated in Section 7.4 are no longer applicable; instead, we can only calculate expectations for individual periods, given the retailers' inventory levels at the start of the period. Dynamic programming is an appropriate tool for solving this problem. Assume the periods are numbered $1, \ldots, T$.

The sequence of events given in Section 7.4.2 applies to this finite-horizon problem as well. Let x_i be the starting inventory level for retailer i in step 1 of the sequence of events, and let y_i be the order-up-to level for retailer i in step 2; that is, $y_i = x_i + Q_i$. Let $\mathbf{x} = (x_1, x_2)$ be the vector of starting inventory levels and $\mathbf{y} = (y_1, y_2)$ be the vector of order-up-to levels.

To keep things simple, assume that retailer 2's demand is deterministic and equal to d_2 in each period. Assume further that retailer 2 always chooses an order-up-to level y_2 that is at least equal to d_2, so that it never experiences stockouts. This also means that transshipments only ever go in one direction: from retailer 2 to retailer 1.

a) Let $g_i(\mathbf{y})$ be the expected holding and stockout costs for retailer i in a given period assuming that $\mathbf{y}$ is the vector of order-up-to levels chosen in step 2. Then

Table A.3 V_{ni} values for Problem A.9.

Souvenir	V_{ni}
Hat	0.48
T-shirt	0.34
Puffy hand	0.21

an expression for $g_2(\mathbf{y})$ is:

$$g_2(\mathbf{y}) = h\left[F_1(y_1)(y_2 - d_2) + \int_{d_1=y_1}^{y_1+(y_2-d_2)} (y_2 - d_2 - (d_1 - y_1))f_1(d_1)dd_1\right].$$

Write an expression for $g_1(\mathbf{y})$.

b) Let $\Gamma(\mathbf{y})$ be the expected transshipment costs in a given period assuming that $\mathbf{y}$ is the vector of order-up-to levels chosen in step 2. Write an expression for $\Gamma(\mathbf{y})$.

c) Let $\theta_t(\mathbf{x})$ be the optimal expected cost in periods $t, \ldots, T$ if retailer i begins period t with inventory level x_i (and each retailer acts optimally thereafter). Write an expression for $\theta_t(\mathbf{x})$. Your expression should make use of the functions defined above, as well as $\theta_{t+1}(\mathbf{x})$.

A.9 (Using Discrete Choice to Forecast Demand) A minor-league baseball stadium has sold 8000 tickets to tonight's baseball game. The stadium sells three kinds of souvenirs: hats, T-shirts, and puffy hands. Each person who attends the game will buy exactly one souvenir. From historical data, the concession manager at the stadium has developed an estimate, V_{ni}, of the utility that each attendee n derives from each of the souvenirs, for $i = 1, 2, 3$. These V_{ni} values are given Table A.3.

a) Assume that the actual utilities U_{ni} differ from the estimated utilities V_{ni} by an additive iid error term that has a standard Gumbel distribution. Using the multinomial logit model, calculate the expected demand for each souvenir.

b) Let X be a random variable representing the total number of people who buy hats. What is the probability distribution of X? Specify the name and the parameters of the distribution.

c) The concession manager replenishes the inventory of hats before the game, and any unsold hats after the game incur an opportunity cost of \$0.25. Unmet demands for hats incur a stockout cost (including lost profit and loss of goodwill) of \$1.75 per hat. How many hats should the manager stock for tonight's game?

Note: You may solve this problem using the discrete demand distribution you identified in part (b), or you may approximate this distribution with a continuous distribution. If you take the latter approach, justify your approximation carefully.

A.10 (Base-Stock Policies with Disruptions) Consider a finite-horizon, periodic-review inventory system with stochastic demand and no fixed cost, for which we proved in Section 4.5.1.2 that a base-stock policy is optimal. There are T periods, the lead time is 0, the demand per period is random with pdf $f(\cdot)$ and cdf $F(\cdot)$, the holding cost is h per item per period, the backorder cost is p per item per period, the per-unit ordering cost is c, and the discount factor is $0 < \gamma \leq 1$. If the inventory level is x at the start of period T, we incur a terminal cost of $\theta_{T+1}(x)$, a convex function.

Now suppose that the supplier is unreliable, and that when an order is placed, the supplier delivers it with probability q ($0 < q \le 1$). With probability $1 - q$, the supplier is disrupted—it's as though the order had never been placed, and the firm must wait until the next time period to order again. Evidently, the order placed in the next time period will be larger to make up for the failed order. (This is a special case of the model in Section 9.2.2 in which disruptions follow an iid Bernoulli process.)

a) Write an expression for $\theta_t(x)$, the expected cost in periods $t, \ldots, T$ if we begin period t with an inventory level equal to x and act optimally in every period. Your expression should be analogous to (4.85) and may use the function $H_t(y)$. If you modify the definition of $H_t(y)$, explain your modifications carefully.

b) Prove that a base-stock policy is optimal in every period t.

Note: If you use any results from Section 4.5.1.2 to prove this, argue why these results are still true under your revised cost functions.

c) Suppose $\theta_{T+1}(x) = -cx$. We know from Section 4.5.1.2 that the same S is optimal in every period if $q = 1$. Do you think the same statement is true if $q < 1$? Explain your answer.

A.11 (**Coordinating the Unreliable Supply Chain**) Consider the newsvendor problem with disruptions discussed in Section 9.2.2. In this problem, you will extend this model to consider the unreliable supplier and the coordination between the two players.

The supplier holds no inventory and has a lead time of 0 when operational. But the supplier is subject to disruptions, and when a disruption occurs, the supplier cannot provide any items. The retailer acts like a newsvendor with deterministic demand of d per period. Excess inventory at the end of the period is held over until the next period at a cost of h_r per unit per period, and unmet demands are backordered, incurring a stockout penalty of p_r per unit per period for the retailer and p_s for the supplier. Let $p \equiv p_r + p_s$.

The supplier's disruption probability is α and its recovery probability is β. The steady-state probability of being in a disruption that has lasted for n periods is π_n, and the cumulative probability of being in a disruption lasting n periods or fewer is $F(n)$, as in (9.10).

Suppose the supplier and retailer have agreed upon a contract that specifies a transfer payment to be made from the retailer to the supplier in an amount based on the current state of the system. From (9.14), the retailer's expected cost can be expressed as a function of its base-stock level S as follows:

$$g_r(S) = h_r \sum_{n=0}^{S/d-1} \pi_n[S - (n+1)d] + p_r \sum_{n=S/d}^{\infty} \pi_n[(n+1)d - S] + T, \tag{A.1}$$

where T is the expected transfer payment. Similarly, the supplier's expected cost is given by

$$g_s(S) = p_s \sum_{n=S/d}^{\infty} \pi_n[(n+1)d - S] - T \tag{A.2}$$

and the total supply chain expected cost is given by

$$G(S) = h_r \sum_{n=0}^{S/d-1} \pi_n[S - (n+1)d] + p \sum_{n=S/d}^{\infty} \pi_n[(n+1)d - S]. \tag{A.3}$$

Let S_r^*, S_s^*, and S^0 be the retailer's, supplier's, and supply chain optimal base-stock level, respectively.

Note that this model is expressed in terms of costs, not profits, and that it assumes backorders, not lost sales; both are changes from the assumptions in Chapter 14.

- **a)** Suppose $T = 0$. Prove that $S_r^* \leq S^0$, and that for some instances, the inequality is strict (in which case the supply chain is not coordinated).
- **b)** In one or two sentences, explain why the retailer tends to under-order in part (a).
- **c)** Now consider a buyback contract in which the retailer pays the supplier w per unit ordered and the supplier pays the retailer b per unit on-hand at the end of each period. (Note that the items remain on-hand; they are not sent back to the supplier or destroyed. They can, rather, be used to satisfy demand in future periods.) Write expressions for $g_r(S, w, b)$, $g_s(S, w, b)$, and $G(S, w, b)$ under this contract.
- **d)** Write the optimal base-stock levels S_r^*, S_s^*, and S^0 under this contract.
- **e)** Find a value for b in terms of the other problem parameters such that $S_r^* = S^0$.
- **f)** Show that, using the b you found in the previous part, $S_s^* = S^0$. Thus, the supply chain is coordinated.
- **g)** You should have found that b and the optimal S values don't depend on w. Explain in one or two sentences why this makes sense.

A.12 (An Approximate Location–Routing Problem) Consider a facility location problem in which the customers assigned to each facility are served via a single truck whose route is determined by solving a traveling salesman problem (TSP). The length of the optimal TSP tour through n points located in an area A is often approximated as $k\sqrt{nA}$, where k is a constant. This approximation is called the "square-root rule." (See Section 10.6.5.)

- **a)** Formulate the problem of locating facilities to minimize the sum of the fixed cost and transportation cost, which is approximated using the square-root rule. Make the (unrealistic) assumption that we know, in advance, that if facility j is opened, then the area of the region it will serve is equal to A_j, where A_j is a parameter of the model.
- **b)** Propose an algorithmic method to solve this problem.

A.13 (A Location–Flexibility Design Problem) We wish to locate facilities and decide which facilities will produce which products. Let P be the set of products. Customer demands are stochastic and are described by scenarios; let S be the set of scenarios and let q_s be the probability that scenario $s \in S$ occurs. There is a fixed cost k_{jp} if facility j is configured to produce product $p \in P$. Facility $j \in J$ has a fixed capacity of v_j, and it takes one unit of capacity to produce one unit of product p, for all $p \in P$. Transportation costs are product-specific. We choose facility locations and capabilities (i.e., assignments of products to facilities) before the scenario is observed, and we set the production quantities, shipment quantities, and assignments of customers to facilities after the scenario is observed.

- **a)** Formulate a linear mixed-integer programming model to minimize the expected cost of this system. Define any new notation clearly. Explain the objective function and each of the constraints in words.
- **b)** Sketch an idea for an algorithm to solve this problem.

A.14 (Flying across the United States) Suppose you would like to fly to each of the 48 continental United States in your private airplane. You don't care where you go in each state; your only requirement is that you land your airplane in one airport per state and then return to the starting airport. Your goal is to minimize the total distance you fly. (The distance does not include the flight from your home airport to the starting airport on your route.)

This problem is an example of the *generalized traveling salesman problem* (GTSP). In the GTSP, the set of nodes is partitioned into subsets, called *clusters*. We must choose exactly one node from each cluster, as well as a route that visits each of the chosen nodes once and returns to the starting node, in order to minimize the total distance traveled. The GTSP therefore combines elements of facility location problems (choosing the nodes) and vehicle routing problems (choosing the route).

Use the following notation. (Do not define any new notation.)

Sets

N = set of nodes

N_r = a cluster, $r = 1, \ldots, R$; the clusters do not overlap and their union equals N

Parameters

c_{jk} = distance between nodes j and k; assume the distances are symmetric ($c_{jk} = c_{kj}$)

Decision Variables

y_i = 1 if node $i \in N$ is selected to be part of the tour, 0 otherwise

x_{ij} = 1 if the tour goes directly from node $i \in N$ to node $j \in N$, 0 otherwise

a) Formulate this problem as a linear discrete optimization problem. Explain the objective function and all constraints in words.

b) Propose a construction heuristic for the GTSP. Explain your heuristic briefly.

c) Propose an improvement heuristic for the GTSP. Explain your heuristic briefly.

APPENDIX B

HOW TO WRITE PROOFS: A SHORT GUIDE

B.1 HOW TO PROVE ANYTHING

OK, fine—we can't actually tell you how to prove *everything*. But we can give you some advice that will help you when you try to prove *anything*.

Writing a proof is more art than science. Although there may be a "correct" way to prove something (or several correct ways), there is still a wide range of styles, formats, and logical implications that follow the same basic argument. (Similarly, you and your friend might write very different essays on "How I Spent My Summer Vacation," even if you did the same things during your vacations.)

Writing a proof is very much like arguing a case in court. (Or at least it's like how it looks on TV.) Like a courtroom argument, a proof should contain a beginning, a middle, and an end.

- The **beginning** tells us what is already known (the assumptions of the theorem), reminds us of important facts that are already in evidence that will be important for the proof, establishes new notation that you will use in the proof, and gives us a hint of where you're headed and what steps the proof will take. Here are some examples:

Courtroom Claim: Dr. Evil is guilty of stealing pencils from Prof. Plum's desk.

Fundamentals of Supply Chain Theory, Second Edition. Lawrence V. Snyder and Zuo-Jun Max Shen.

Companion website: www.wiley.com/go/Snyder/SupplyChainTheory

> **Courtroom Argument:** Consider the man sitting before you, Dr. Evil. You already know that security camera video from the night of October 6 shows Dr. Evil entering Prof. Plum's office building. Let the security video tape from that night be labeled as "Exhibit A." Today I will convince you, beyond a reasonable doubt, that Dr. Evil stole pencils from Prof. Plum's desk on that night. I will do this by providing physical evidence placing Dr. Evil in Prof. Plum's office and demonstrating that Dr. Evil had, indeed, touched pencils recently.

> **Theorem B.1** *The sum of two convex functions is also convex.*
> **Proof:** Consider two convex functions $f(x)$ and $g(x)$. Recall that, by definition, a function $f(x)$ is convex if, for any x and y in its domain and for any $\lambda \in [0, 1]$,
>
> $$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y). \tag{B.1}$$
>
> Let $h(x) \equiv f(x)+g(x)$, let x and y be in the domain of f and g, and let $\lambda \in [0, 1]$. We will show that $h(x)$ is convex by proving that inequality (B.1) holds.

See how similar their structures are?

- The **middle** provides the evidence that proves the claim. Like evidence in a trial, the steps in your proof must follow logically from one another and must be straightforward to follow.

> **Courtroom Argument:** Exhibit A shows Dr. Evil entering Prof. Plum's office building at 11:37 PM. At approximately 11:45 PM, graduate students saw a secretive figure attempting to pick the lock on Prof. Plum's office door. Dr. Evil's fingerprints were found on the door, and in Prof. Plum's office, on the following day (October 7), and since Prof. Plum's office had been steam-cleaned the day before, the fingerprints must have been left on the night of October 6. Moreover, graphite stains on Dr. Evil's shirt match the precise composition of graphite contained in the pencils that Prof. Plum keeps in his desk.

> **Proof:**
>
> $$\begin{aligned} h(\lambda x + (1-\lambda)y) &= f(\lambda x + (1-\lambda)y) + g(\lambda x + (1-\lambda)y) \\ &\leq [\lambda f(x) + (1-\lambda)f(y)] + [\lambda g(x) + (1-\lambda)g(y)] \\ &= \lambda[f(x) + g(x)] + (1-\lambda)[f(y) + g(y)] \\ &= \lambda h(x) + (1-\lambda)h(y) \end{aligned}$$
>
> The first and last equalities follow from the definition of $h(x)$. The inequality follows from the fact that $f(x)$ and $g(x)$ are convex functions.

Note that, in the proof, we provided justification for the steps that were not immediately obvious but omitted justification for the easy algebraic step. What counts as "easy" or "obvious" is, of course, a subjective matter. In general, a good rule of thumb to use is that, if your fellow students were reading your proof, *they should be able to follow each step without having to look up any facts or write down any additional derivations.*

- The **end** is the arrival point—the claim you are trying to prove.

> **Courtroom Argument:** Ladies and gentlemen of the jury, the preponderance of evidence demonstrates that Dr. Evil has committed this heinous crime. You have no choice but to find him guilty.

Proof: Therefore, $h(x)$ is convex. ■

Just like the lawyer's argument, the proof uses words—not just math—to lead the reader on the path from assumptions to conclusions. Of course, in a legal trial, facts and implications are subject to interpretation—that's why we have judges and juries. In a mathematical proof, however, all facts and logical implications should be incontrovertible.

B.2 TYPES OF THINGS YOU MAY BE ASKED TO PROVE

Here is a (very nonexhaustive) list of the *kinds* of statements you may be asked to prove in this book or at some other point in your proof-writing career:

- $x = y$

  This is the simplest kind of statement (though that does not mean it will require the simplest proof). It simply asks you to prove that two mathematical objects are equal.

  Theorem B.2 *Let* $g(Q) = \frac{K\lambda}{Q} + \frac{hQ}{2}$ *and* $Q^* = \sqrt{2K\lambda/h}$. *Then* $g(Q^*) = \sqrt{2K\lambda h}$.

  Notice that in this example, there is a qualifying statement to set up the statement you are asked to prove; this is fairly typical.

- $p \implies q$

  The symbols p and q stand here not for variables but for statements. The symbol $\implies$ is interpreted as "p implies q," and it is the same as saying "if p then q."

  Theorem B.3 *If* $0 < r < 1$, *then*

  $$\sum_{n=0}^{\infty} r^n = \frac{1}{1-r}. \tag{B.2}$$

  Here p is "$0 < r < 1$" and q is equation (B.2).

- $p \iff q$

  The symbol $\iff$ means "if and only if" (sometimes abbreviated "iff"). The claim indicates that either both statements are true or both are false. Another description for this kind of statement is that q is a necessary and sufficient condition for p—in order for p to be true, q must be true, and the truth of q is sufficient to ensure the truth of p.

  Theorem B.4 *In the newsvendor model under normally distributed demand, the optimal safety stock level,* $\sigma\Phi^{-1}\left(\frac{p}{p+h}\right)$, *is positive if and only if* $p > h$.

  To prove an iff statement, you must prove both directions of the implication—that is, you must prove that p implies q and that q implies p. Sometimes you can do this all at once using a string of iff implications:

  $$\sigma\Phi^{-1}\left(\frac{p}{p+h}\right) > 0$$

$$\iff \Phi^{-1}\left(\frac{p}{p+h}\right) > 0$$
$$\iff \frac{p}{p+h} > \frac{1}{2}$$
$$\iff p > h$$

In other cases, though, the proof needs to be divided into two parts. In the first part, you prove one implication (e.g., $p \implies q$), and in the second part, you prove the reverse implication ($q \implies p$, or its logical equivalent, $\neg p \implies \neg q$).

- $\forall x$ such that [*condition*], [*statement*].

  Here you are asked to prove that for all x that satisfy a certain [*condition*], some [*statement*] is true.

  Theorem B.5 *For all x such that $x \geq 1$, $\ln x \geq 0$.*

  To prove a "$\forall x$" claim, you take the [*condition*] as given and prove that the [*statement*] is true. This actually feels a lot like a "$p \implies q$" claim, and in fact they are often logically equivalent.

- $\exists x$ such that [*statement*].

  This time you need to prove that there exists (at least one) x that satisfies the [*statement*]. Sometimes there are qualifying conditions on the type of x that are allowed.

  Theorem B.6 *Suppose that $\lim_{x\to\infty} f(x) = \infty$. Then for any x', there exists an $x > x'$ such that $f(x) > f(x')$.*

  (In this case, there are many x that will do the trick, but you are asked only to prove the existence of one of them.)

- $\neg p$

  In other words, you are being asked to *disprove* the statement p.

  Theorem B.7 *Let $f(x)$ and $g(x)$ be convex functions and let $h(x) \equiv f(x)g(x)$. Then $h(x)$ is not necessarily convex.*

  In general, it suffices to find a single example for which p is not true. In the example above, you just need to find two convex functions whose product is not convex.

  However, if p is of the form "$\exists x$ such that [*statement*]," then to disprove p you must prove that the [*statement*] is false *for all x*. This may be easy or hard. Here's a hard example:

  Theorem B.8 *Let n be an integer greater than 2. Then there do not exist integers a, b, and c such that*
  $$a^n + b^n = c^n.$$

  (This is Fermat's Last Theorem, which went unproved for over 350 years until it was finally proved in 1995.)

B.3 PROOF TECHNIQUES

This section will give you a quick overview of several types of proofs—strategies for proving theorems. (Of course, there are others that this list does not include.) These are tools in your proof-building toolbox. It's your job to figure out which tool(s) to use for each job.

B.3.1 Direct Proof

This is the most common kind of proof—you simply prove the claim directly, through a series of logical implications.

Theorem B.9 *If* $0 < r < 1$*, then*

$$\sum_{n=0}^{\infty} r^n = \frac{1}{1-r}.$$

Proof. Let $0 < r < 1$. Define $A \equiv \sum_{n=0}^{\infty} r^n$. We wish to prove that $A = \frac{1}{1-r}$. Well,

$$\begin{aligned} A &= r^0 + \sum_{n=1}^{\infty} r^n \\ &= 1 + r\sum_{n=0}^{\infty} r^n \\ &= 1 + rA. \end{aligned}$$

Therefore,

$$A = \frac{1}{1-r},$$

as desired. ■

B.3.2 Proof by Contradiction

Suppose you are trying to prove that $p \implies q$. In a proof by contradiction, you assume p, as usual, but then you assume that q is *not* true and then prove that a contradiction occurs. In particular, you show that if q is false, then so is p. And since you have assumed that p is true, you have now proven that p is both true and false—an impossibility. Therefore, the assumption of $\neg q$ must have been false—in other words, q must be true.

Theorem B.10 *There are an infinite number of prime numbers.*

Actually, to highlight the structure of a proof by contradiction, let's rewrite the theorem in "$p \implies q$" form, even though it's a little more awkward:

Theorem B.11 (Theorem B.10 Revised) *If* N *is the number of prime numbers, then* $N = \infty$.

Proof. Suppose (for a contradiction) that N is finite. Let the N primes be denoted $p_1, p_2, \ldots, p_N$. Furthermore, let

$$B = \prod_{n=1}^{N} p_n + 1.$$

Now, B is also a prime number: None of the primes $p_1, \ldots, p_N$ divides B (each results in a remainder of 1), so by definition, B is prime. Moreover, B is not in $\{p_1, \ldots, p_N\}$ since it is larger than each of the p_n. Therefore, we have found a new prime, so there must be at least $N + 1$ of them, contradicting our assumption that the number of primes is N. Therefore, there are an infinite number of primes. ■

Note the parenthetical phrase "for a contradiction." This is not strictly necessary, but it does help the reader by letting him or her know that the assumption you're about to make is not one that you actually believe—you are making it solely for the purpose of proving a contradiction later.

The *contrapositive* of the statement $p \implies q$ is $\neg q \implies \neg p$, and the two are logically equivalent; therefore, you can prove $p \implies q$ by proving $\neg q \implies \neg p$. This feels a lot like a proof by contradiction—we assume $\neg q$ and prove $\neg p$. The difference is that in a proof by contradiction, we *also assume* p and we use it to derive the contradiction. For example, we used the fact that N is the number of primes to build B, and then we used B to contradict the fact that N is the number of primes. In a proof by contrapositive, we don't need to assume p—we simply assume $\neg q$ and prove $\neg p$.

B.3.3 Proof by Mathematical Induction

Mathematical induction is useful when you need to prove something about all the integers (or all the members of some other countable set). The idea is to prove that if the claim is true for (an arbitrary) n, then it must also be true for $n+1$. If we can prove this implication, then it holds for any n—that is, the truth of the claim for n implies the claim for $n + 1$, and this in turn implies the claim for $n + 2$, and then for $n + 3$, and so on. Therefore, this general implication (truth for $n \implies$ truth for $n + 1$) is powerful enough to prove that the claim is true for all integers greater than or equal to n. Of course, we also have to get the process started, by proving that the claim is true for $n = 1$.

A proof by induction generally has two parts: In the first (often called the *base case*), we prove that the claim is true for $n = 1$, and in the second (called the *induction step*), we prove that, if the claim holds for n, then it also holds for $n + 1$. In the induction step, we are allowed to assume that the claim holds for n—this is called the *induction hypothesis*.

Theorem B.12 *For all integers $n \geq 1$,*

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}.$$

Proof: By induction on n.

<u>Base Case</u>: If $n = 1$, then

$$\sum_{i=1}^{n} i = 1 = \frac{1(2)}{2}.$$

<u>Induction Step</u>: Suppose the claim holds for n. We need to prove that the claim holds for $n + 1$, i.e., that

$$\sum_{i=1}^{n+1} i = \frac{(n+1)(n+2)}{2}.$$

Well,

$$\begin{aligned}\sum_{i=1}^{n+1} i &= \sum_{i=1}^{n} i + (n+1)\\ &= \frac{n(n+1)}{2} + (n+1) \qquad \text{(by the induction hypothesis)}\\ &= \frac{n(n+1)+2(n+1)}{2}\\ &= \frac{(n+1)(n+2)}{2},\end{aligned}$$

as desired. ∎

Note the phrase "by induction on n" at the start of the proof. This is not strictly necessary, but it helps the reader by telling him or her how we're going to prove the theorem.

B.3.4 Proof by Cases

In this method, the universe of possibilities is divided into cases, and the claim is proved for each case separately. We don't know which case applies—typically, all of the cases are possible. But since we've proved the claim for every case, it doesn't matter which case holds.

Theorem B.13 *Let k be a perfect cube. Then k is either a multiple of 9, or 1 more than a multiple of 9, or 1 less than a multiple of 9.*

Proof. Since k is a perfect cube, there exists an integer n such that $k = n^3$. Every integer is either a multiple of 3, or 1 more than a multiple of 3, or 1 less than a multiple of 3. We will consider three cases:

Case 1: n is a multiple of 3.

Then there exists an integer p such that $n = 3p$. Then $k = n^3 = 27p^3$, so k is a multiple of 9.

Case 2: n is 1 more than a multiple of 3.

Then there exists an integer p such that $n = 3p + 1$. Then $k = n^3 = 27p^3 + 27p^2 + 9p + 1$, which is 1 more than a multiple of 9.

Case 3: n is 1 less than a multiple of 3.

Then there exists an integer p such that $n = 3p - 1$. Then $k = n^3 = 27p^3 - 27p^2 + 9p - 1$, which is 1 less than a multiple of 9. ∎

B.4 OTHER ADVICE

- *Provide explanations, not just math.* Even though your reader may be a smart mathematician, you should still provide verbal explanations that explain the intuition behind the math whenever the math is a bit complicated. This applies to derivations, but also to definitions.

  For example, suppose you want to say:

  > Let t_i be the order times, let T be the set of order times, let $S = \{t|I(t^-) > 0, t \in T\}$, and let $t_{\min} = \text{argmin}_t\{t \in S\}$.

Then you may wish to help out your reader a bit by also saying:

> That is, S is the set of order times for which the inventory level (just before ordering) is positive, and $t_{\min}$ is the earliest such time.

- *Distinguish between definitional and derivational =’s.* The = sign has two meanings: One means “let the left-hand side be defined to equal the right-hand side” and the other means “I have now proved that the left-hand side equals the right-hand side.” The first is a definitional equality, the second is a derivational equality. It is important to differentiate between them. The best way to do this is using words: “Let $x = y^2/2$.” “Therefore, $s = r - D$.” (The difference between these two types of equality is exactly the same as the difference between = and == in C/C++, Java, and other programming languages. It’s also the difference between $\leftarrow$ and $=$ in the pseudocode in this book.)

  For example, consider the following proof fragment:

$$
\begin{aligned}
v &= r - D \\
y &= r^2 \\
D &= 2y \\
v &= \sqrt{y} - 2y
\end{aligned}
$$

  This fragment is confusing. Is v a new symbol that is being defined as $r - D$? Or do we already know that $v = r - D$? Did the first step prove that $y = r^2$? Or do we already know that $y = r^2$? Or is it another new symbol?

  True, a smart reader might be able to figure all this out. But the reader’s life would be a lot easer if the proof-writer instead wrote:

> We know that $v = r - D$. Let $y = r^2$. Since $D = 2y$ (by Theorem 4.3), we have
> $$v = r - D = \sqrt{y} - 2y.$$

  This bullet is really a special case of the next.

- *Use complete sentences.* The first proof fragment in the previous bullet (starting with $v = r - D$) becomes a lot easier to read when it’s written using complete sentences. If you use sentences, the ambiguities of the = sign are resolved. The same could be said about many other mathematical ambiguities and confusions. Writing in complete sentences—even if your English isn’t very good—will instantly make your proofs easier to read.

- *Typeset thoughtfully.* If you are writing your proofs by hand, take care to write them neatly, and think carefully about how the proof will be laid out, including your use of white space. Even better, type your proofs using LaTeX or another software package for typesetting mathematical text. Invest the time to learn how to typeset complicated math so that it looks nice and helps convey your meaning. Be considerate of your reader.

  For example, consider the following proof fragment:

$$[(hQ/2) + (K\lambda/Q)]/(hQ^*) = [hQ^2 + 2K\lambda]/(2hQQ^*).$$

The math would be a lot easier to follow if the proof-writer had written the fractions the way they were intended to be written:

$$\frac{\frac{hQ}{2} + \frac{K\lambda}{Q}}{hQ^*} = \frac{hQ^2 + 2K\lambda}{2hQQ^*}.$$

In general, thoughtfully typeset math will make your proof easier to read.

- *Don't stop here.* There are many books and other resources for learning how to write proofs. (See, e.g., Sundstrom (2006) and Velleman (2006).) There are also lots of web sites devoted to the topic. Like web sites devoted to any topic, some of these are very good and others are very bad, so be a good critic when you read.

APPENDIX C

HELPFUL FORMULAS

C.1 POSITIVE AND NEGATIVE PARTS

For any number $x \in \mathbb{R}$, we define the *positive part* and *negative part* of x as, respectively:

$$x^+ = \max\{x, 0\} = x \text{ if } x > 0 \text{ and } 0 \text{ otherwise}$$
$$x^- = |\min\{x, 0\}| = |x| \text{ if } x < 0 \text{ and } 0 \text{ otherwise}$$

(Some authors use $x^- = \min\{x, 0\}$.)

The following identities hold:

$$x = x^+ - x^- \tag{C.1}$$
$$|x| = x^+ + x^- \tag{C.2}$$
$$(-x)^+ = x^- \tag{C.3}$$
$$(-x)^- = x^+ \tag{C.4}$$

For any $x, y \in \mathbb{R}$, we have:

$$\min\{x, y\} = x - (x - y)^+ = y - (y - x)^+ \tag{C.5}$$
$$\max\{x, y\} = x + (y - x)^+ = y + (x - y)^+ \tag{C.6}$$

If $y \geq 0$, then

$$(x^+ - y)^+ = (x - y)^+. \tag{C.7}$$

Fundamentals of Supply Chain Theory, Second Edition. Lawrence V. Snyder and Zuo-Jun Max Shen.

Companion website: www.wiley.com/go/Snyder/SupplyChainTheory

C.2 STANDARD NORMAL RANDOM VARIABLES

Let $X \sim N(\mu, \sigma^2)$ with pdf f and cdf F. Let ϕ and Φ be the pdf and cdf, respectively, of the standard normal distribution.

$$f(x) = \frac{1}{\sigma}\phi\left(\frac{x-\mu}{\sigma}\right) \tag{C.8}$$

$$F(x) = \Phi\left(\frac{x-\mu}{\sigma}\right) \tag{C.9}$$

We define

$$z_\alpha = \Phi^{-1}(\alpha) \tag{C.10}$$

for $0 < \alpha < 1$. Moreover,

$$z_\alpha = -z_{1-\alpha}. \tag{C.11}$$

C.3 LOSS FUNCTIONS

Throughout, we use $n(\cdot)$ and $n^{(2)}(\cdot)$ to refer to the first- and second-order *loss functions*, and $\bar{n}(\cdot)$ and $\bar{n}^{(2)}(\cdot)$ to refer to the corresponding *complementary loss functions*.[1] It would be equally appropriate to use $n^{(1)}(\cdot)$ for the first-order loss function, but we drop the superscript for notational simplicity, and often omit the phrase "first-order" when describing this function and its complement. For the standard normal distribution, we replace n with $\mathscr{L}$ in these functions.

C.3.1 General Continuous Distributions

Let X be a continuous random variable with pdf f and cdf F. Let $\bar{F}(x) = 1 - F(x)$ be the complementary cdf. The loss function and complementary loss function are given by

$$n(x) = \mathbb{E}[(X-x)^+] = \int_x^\infty (y-x)f(y)dy = \int_x^\infty \bar{F}(y)dy \tag{C.12}$$

$$\bar{n}(x) = \mathbb{E}[(X-x)^-] = \int_{-\infty}^x (x-y)f(y)dy = \int_{-\infty}^x F(y)dy. \tag{C.13}$$

The loss function and its complement are related as follows:

$$\bar{n}(x) = x - \mathbb{E}[X] + n(x). \tag{C.14}$$

The derivatives of the loss function and its complement are given by

$$n'(x) = F(x) - 1 \tag{C.15}$$

$$\bar{n}'(x) = F(x). \tag{C.16}$$

The loss function and its complement are therefore both convex.

[1] The term "complementary loss function" and the notation $\bar{n}(x)$ are our own. They are not standard.

The second-order loss function and its complement are given by

$$n^{(2)}(x) = \frac{1}{2}\mathbb{E}\left[\left([X-x]^+\right)^2\right] = \frac{1}{2}\int_x^\infty (y-x)^2 f(y)dy = \int_x^\infty n(y)dy \qquad \text{(C.17)}$$

$$\bar{n}^{(2)}(x) = \frac{1}{2}\mathbb{E}\left[\left([X-x]^-\right)^2\right] = \frac{1}{2}\int_{-\infty}^x (x-y)^2 f(y)dy = \int_{-\infty}^x \bar{n}(y)dy. \qquad \text{(C.18)}$$

The second-order loss function and its complement are related as follows:

$$\bar{n}^{(2)}(x) = \frac{1}{2}\left((x-\mathbb{E}[X])^2 + \mathrm{Var}[X]\right) - n^{(2)}(x). \qquad \text{(C.19)}$$

The derivatives of the second-order loss function and its complement are given by

$$\frac{d}{dx}n^{(2)}(x) = -n(x) \qquad \text{(C.20)}$$

$$\frac{d}{dx}\bar{n}^{(2)}(x) = \bar{n}(x). \qquad \text{(C.21)}$$

C.3.2 Standard Normal Distribution

Let $Z \sim N(0,1)$, with pdf ϕ, cdf Φ, and complementary cdf $\bar{\Phi}$. The standard normal loss function, its complement, and their derivatives are given by

$$\mathscr{L}(z) = \mathbb{E}[(Z-z)^+] = \int_z^\infty (t-z)\phi(t)dt = \phi(z) - z\bar{\Phi}(z) \qquad \text{(C.22)}$$

$$\bar{\mathscr{L}}(z) = \mathbb{E}[(Z-z)^-] = \int_{-\infty}^z (z-t)\phi(t)dt = z + \mathscr{L}(z) \qquad \text{(C.23)}$$

$$\mathscr{L}'(z) = \Phi(z) - 1 \qquad \text{(C.24)}$$

$$\bar{\mathscr{L}}'(z) = \Phi(z). \qquad \text{(C.25)}$$

Also:

$$\mathscr{L}(-z) = z + \mathscr{L}(z) = \bar{\mathscr{L}}(z). \qquad \text{(C.26)}$$

(The second equality follows from the fact that $\mathbb{E}[Z] = 0$.)

The second-order standard normal loss function, its complement, and their derivatives are given by

$$\begin{aligned}\mathscr{L}^{(2)}(z) &= \frac{1}{2}\mathbb{E}\left[\left([Z-z]^+\right)^2\right] = \frac{1}{2}\int_z^\infty (t-z)^2\phi(t)dt \\ &= \frac{1}{2}\left[\left(z^2+1\right)\bar{\Phi}(z) - z\phi(z)\right] \qquad \text{(C.27)}\end{aligned}$$

$$\begin{aligned}\bar{\mathscr{L}}^{(2)}(z) &= \frac{1}{2}\mathbb{E}\left[\left([Z-z]^-\right)^2\right] = \frac{1}{2}\int_{-\infty}^z (z-t)^2 f(t)dt \\ &= \frac{1}{2}(z^2+1) - \mathscr{L}^{(2)}(z). \qquad \text{(C.28)}\end{aligned}$$

$$\frac{d}{dz}\mathscr{L}^{(2)}(z) = -\mathscr{L}(z) \qquad \text{(C.29)}$$

$$\frac{d}{dz}\bar{\mathscr{L}}^{(2)}(z) = \bar{\mathscr{L}}(z). \qquad \text{(C.30)}$$

C.3.3 Nonstandard Normal Distributions

Let $X \sim N(\mu, \sigma^2)$ with pdf f, cdf F, and complementary cdf $\bar{F}$. The normal loss function can be computed using the standard normal loss function as follows:

$$n(x) = \int_x^\infty (y-x)f(y)dy = \mathscr{L}(z)\sigma \tag{C.31}$$

$$\bar{n}(x) = \int_{-\infty}^x (x-y)f(y)dy = \bar{\mathscr{L}}(z)\sigma, \tag{C.32}$$

where $z = (x-\mu)/\sigma$. (In many instances, we assume $\sigma \ll \mu$ so that the probability that $X < 0$ is small; in these cases, we often replace the lower limit of the integral in (C.32) with 0.) The derivatives of $n(x)$ and $\bar{n}(x)$ are given by (C.15)–(C.16).

The second-order normal loss function and its complement are given by

$$n^{(2)}(x) = \frac{1}{2}\int_x^\infty (y-x)^2 f(y)dy = \mathscr{L}^{(2)}(z)\sigma^2 \tag{C.33}$$

$$\bar{n}^{(2)}(x) = \frac{1}{2}\int_{-\infty}^x (x-t)^2 f(t)dt = \bar{\mathscr{L}}^{(2)}(z)\sigma^2. \tag{C.34}$$

The derivatives of $n^{(2)}(x)$ and $\bar{n}^{(2)}(x)$ are given by (C.20)–(C.21).

C.3.4 General Discrete Distributions

Let X be a discrete random variable with pmf f and cdf F. Let $\bar{F}(x) = 1 - F(x)$ be the complementary cdf. The loss function and complementary loss function are given by

$$n(x) = \mathbb{E}[(X-x)^+] = \sum_{y=x}^\infty (y-x)f(y) = \sum_{y=x}^\infty \bar{F}(y) \tag{C.35}$$

$$\bar{n}(x) = \mathbb{E}[(X-x)^-] = \sum_{y=-\infty}^{x} (x-y)f(y) = \sum_{y=-\infty}^{x-1} F(y). \tag{C.36}$$

The loss function and its complement are related as follows:

$$\bar{n}(x) = x - \mathbb{E}[X] + n(x). \tag{C.37}$$

The second-order loss function and its complement are given by

$$\begin{aligned} n^{(2)}(x) &= \frac{1}{2}\mathbb{E}\left[(X-x)^+(X-x-1)^+\right] = \frac{1}{2}\sum_{y=x}^\infty (y-x)(y-x-1)f(y) \\ &= \sum_{y=x}^\infty (y-x)\bar{F}(y) = \sum_{y=x+1}^\infty n(y) \end{aligned} \tag{C.38}$$

$$\begin{aligned} \bar{n}^{(2)}(x) &= \frac{1}{2}\mathbb{E}\left[(X-x)^-(X-x-1)^-\right] = \frac{1}{2}\sum_{y=-\infty}^{x} (x-y)(x+1-y)f(y) \\ &= \sum_{y=-\infty}^{x} (x-y)F(y) = \sum_{y=-\infty}^{x} \bar{n}(y). \end{aligned} \tag{C.39}$$

The second-order loss function and its complement are related as follows:

$$\bar{n}^{(2)}(x) = \frac{1}{2}\Big((x - \mathbb{E}[X])^2 + (x - \mathbb{E}[X]) + \mathrm{Var}[X]\Big) - n^{(2)}(x). \tag{C.40}$$

If X is nonnegative, then equations (C.37) and (C.40) can facilitate the calculation of $n(x)$ and $n^{(2)}(x)$, since $n(x)$ and $n^{(2)}(x)$ contain infinite sums, but $\bar{n}(x)$ and $\bar{n}^{(2)}(x)$ contain finite ones.

C.3.5 Poisson Distribution

Let $X \sim \mathrm{Pois}(\lambda)$ with pmf f, cdf F, and complementary cdf $\bar{F}$. The Poisson loss function and complementary loss function are given by

$$n(x) = \mathbb{E}[(X - x)^+] = \sum_{y=x}^{\infty}(y - x)f(y) = -(x - \lambda)\bar{F}(x) + \lambda f(x) \tag{C.41}$$

$$\bar{n}(x) = \mathbb{E}[(X - x)^-] = \sum_{y=-\infty}^{x}(x - y)f(y) = (x - \lambda)F(x) + \lambda f(x). \tag{C.42}$$

The second-order Poisson loss function and its complement are given by

$$\begin{aligned} n^{(2)}(x) &= \frac{1}{2}\mathbb{E}\left[[X - x]^+[X - x - 1]^+\right] \\ &= \frac{1}{2}\left[\left[(x - \lambda)^2 + x\right]\bar{F}(x) - \lambda(x - \lambda)f(x)\right] \end{aligned} \tag{C.43}$$

$$\begin{aligned} \bar{n}^{(2)}(x) &= \frac{1}{2}\mathbb{E}\left[[X - x]^-[X - x - 1]^-\right] \\ &= \frac{1}{2}\left[\left[(x - \lambda)^2 + x\right]F(x) + \lambda(x - \lambda)f(x)\right]. \end{aligned} \tag{C.44}$$

C.4 DIFFERENTIATION OF INTEGRALS

C.4.1 Variable of Differentiation Not in Integral Limits

$$\frac{d}{dx}\int_a^b f(t, x)dt = \int_a^b \frac{\partial f(t, x)}{\partial x}dt \tag{C.45}$$

C.4.2 Variable of Differentiation in Integral Limits

$$\frac{d}{dx}\int_a^x f(t)dt = f(x) \tag{C.46}$$

$$\frac{d}{dx}\int_a^{g(x)} f(t)dt = f(g(x))g'(x) \tag{C.47}$$

$$\frac{d}{dx}\int_{g_1(x)}^{g_2(x)} f(t)dt = f(g_2(x))g_2'(x) - f(g_1(x))g_1'(x) \tag{C.48}$$

$$\frac{d}{dx}\int_{g_1(x)}^{g_2(x)} f(t,x)dt = \int_{g_1(x)}^{g_2(x)} \frac{\partial f(t,x)}{\partial x}dt + f(g_2(x),x)g_2'(x) - f(g_1(x),x)g_1'(x) \tag{C.49}$$

Equation (C.49) is known as *Leibniz's rule*.

C.5 GEOMETRIC SERIES

If $0 < |r| < 1$, then:

$$\sum_{i=0}^{\infty} r^i = \frac{1}{1-r} \tag{C.50}$$

$$\sum_{i=0}^{k} r^i = \frac{1-r^{k+1}}{1-r} \tag{C.51}$$

$$\sum_{i=k}^{\infty} r^i = \frac{r^k}{1-r} \tag{C.52}$$

$$\sum_{i=1}^{\infty} ir^{i-1} = \frac{1}{(1-r)^2} \tag{C.53}$$

$$\sum_{i=1}^{k} ir^{i-1} = \frac{1-r^k}{(1-r)^2} - \frac{kr^k}{1-r} \tag{C.54}$$

$$\sum_{i=k}^{\infty} ir^{i-1} = \frac{r^{k-1}}{(1-r)^2} + \frac{(k-1)r^{k-1}}{1-r} \tag{C.55}$$

C.6 NORMAL DISTRIBUTIONS IN EXCEL AND MATLAB

Microsoft Excel and MATLAB have several built-in functions for computing normal distributions. Let $X \sim N(\mu, \sigma^2)$ with pdf f and cdf F and $Z \sim N(0,1)$ with pdf ϕ and cdf Φ. Then, in Excel:

$$\texttt{NORM.DIST}(x,\mu,\sigma,\texttt{cumulative} = \texttt{FALSE}) = f(x) \tag{C.56}$$

$$\texttt{NORM.DIST}(x,\mu,\sigma,\texttt{cumulative} = \texttt{TRUE}) = F(x) \tag{C.57}$$

$$\texttt{NORM.S.DIST}(z,\texttt{cumulative} = \texttt{FALSE}) = \phi(z) \tag{C.58}$$

$$\texttt{NORM.S.DIST}(z,\texttt{cumulative} = \texttt{TRUE}) = \Phi(z) \tag{C.59}$$

$$\texttt{NORM.INV}(p,\mu,\sigma) = F^{-1}(p) \tag{C.60}$$

$$\texttt{NORM.S.INV}(p) = \Phi^{-1}(p) \tag{C.61}$$

And, in MATLAB:

$$\texttt{normpdf}(x,\mu,\sigma) = f(x) \tag{C.62}$$

$$\texttt{normcdf}(x,\mu,\sigma) = F(x) \tag{C.63}$$

$$\texttt{normpdf}(z) = \phi(z) \tag{C.64}$$
$$\texttt{normcdf}(z) = \Phi(z) \tag{C.65}$$
$$\texttt{norminv}(p, \mu, \sigma) = F^{-1}(p) \tag{C.66}$$
$$\texttt{norminv}(p) = \Phi^{-1}(p) \tag{C.67}$$

C.7 PARTIAL EXPECTATIONS

The following formulas computes *partial expectations* of a random variable with pdf f and cdf F. (If $a = -\infty$ and $b = \infty$, these each equal the true mean.)

$$\int_{-\infty}^{b} yf(y)dy = bF(b) - \bar{n}(b) \tag{C.68}$$
$$\int_{a}^{\infty} yf(y)dy = a\bar{F}(a) + n(a) \tag{C.69}$$
$$\int_{a}^{b} yf(y)dy = \bar{n}(a) - \bar{n}(b) - aF(a) + bF(b) \tag{C.70}$$

Discrete versions are also available:

$$\sum_{y=-\infty}^{b} yf(y) = bF(b) - \bar{n}(b) \tag{C.71}$$
$$\sum_{y=a}^{\infty} yf(y) = a\bar{F}(a-1) + n(a) \tag{C.72}$$
$$\sum_{y=a}^{b} yf(y) = \bar{n}(a) - \bar{n}(b) - aF(a-1) + bF(b) \tag{C.73}$$

For a continuous random variable X and constants a and b, the identities above can be used to prove:

$$\mathbb{E}[\min\{aX, b\}] = b - a\bar{n}\left(\frac{b}{a}\right) \tag{C.74}$$
$$\mathbb{E}[\max\{aX, b\}] = b + an\left(\frac{b}{a}\right) \tag{C.75}$$

APPENDIX D
INTEGER OPTIMIZATION TECHNIQUES

In this appendix, we provide a brief overview of two optimization techniques that are used repeatedly in this book: Lagrangian relaxation and column generation.

D.1 LAGRANGIAN RELAXATION

D.1.1 Overview

Consider an optimization problem of the form

$$\begin{aligned}
\text{(P)} \quad \text{minimize} \quad & cx && \text{(D.1)}\\
\text{subject to} \quad & Ax = b && \text{(D.2)}\\
& Dx \le e && \text{(D.3)}\\
& x \ge 0 \text{ and binary} && \text{(D.4)}
\end{aligned}$$

Here, x is a vector of decision variables, b, c, and e are vectors of coefficients, and A and D are matrices. (It's not necessary that all of the x variables be binary; some or all can be continuous.) Suppose that (P) itself is hard to solve, but that the problem obtained by omitting constraints (D.2) is easier. In this section, we discuss Lagrangian relaxation, a method that is well suited to solve problems like this one. Similar approaches can also be applied to other types of problems, such as nonlinear programming problems.

Fundamentals of Supply Chain Theory, Second Edition. Lawrence V. Snyder and Zuo-Jun Max Shen.

Companion website: www.wiley.com/go/Snyder/SupplyChainTheory

There are many sources of additional information about Lagrangian relaxation in journal articles and textbooks; among the most user-friendly treatments are the articles by Fisher (1981, 1985).

The idea behind *Lagrangian relaxation* is to relax (i.e., remove) the hard constraints (D.2) to produce an easier problem. When we remove the constraints, we add a term to the objective function that penalizes solutions for violating the relaxed constraints. This penalty term uses a vector λ of *Lagrange multipliers*, one per constraint, that dictate the magnitude of the penalty. The *Lagrangian subproblem* is then given by

$$
\begin{aligned}
(\text{P-LR}_\lambda) \quad \text{minimize} \quad & cx + \lambda(b - Ax) && \text{(D.5)} \\
\text{subject to} \quad & Dx \le e && \text{(D.6)} \\
& x \ge 0 \text{ and binary} && \text{(D.7)}
\end{aligned}
$$

Problem (P-LR_λ) is easier to solve than problem (P). This, by itself, does not help us very much, because solutions to (P-LR_λ) will typically be infeasible for (P). But it turns out that the optimal solution to (P-LR_λ) provides us with a *lower bound* on the optimal objective value of (P). Feasible solutions to (P) each provide an *upper bound* on the optimal objective value. Such solutions must be found using some other method, typically using a heuristic that is executed once per iteration of the Lagrangian relaxation procedure. When the upper and lower bounds are close (say, within 0.1%), we know that the feasible solution we have found is close to optimal.

When choosing which constraints to relax, i.e., which constraints to label as "hard," there are three main considerations:

- How easy the relaxed problem is to solve
- How tight the resulting lower bound is
- How many constraints are being relaxed

Choosing which constraints to relax is not straightforward, and often some trial and error is required.

D.1.2 Bounds

Let z^* be the optimal objective value of (P) and let $z_{\text{LR}}(\lambda)$ be the optimal objective value of (P-LR_λ). Let m be the number of rows in A, that is, the number of constraints in (D.2). Then we have the following result:

Theorem D.1 *For any $\lambda \in \mathbb{R}^m$,*

$$z_{\text{LR}}(\lambda) \le z^*.$$

Proof. Let x be a feasible solution for (P). Clearly x is feasible for (P-LR_λ), and it has the same objective value in both problems since the constraint violations all equal 0. Therefore, the optimal objective value for (P-LR_λ) is no greater than that of (P). ■

If (P) has a different structure than given in (D.1)–(D.4)—for example, if it is a maximization problem, or if (D.2) are inequality constraints—then we must make some modifications to Theorem D.1 (and the results that follow); see Section D.1.5.

Since (P) is a minimization problem, we want lower bounds that are as large as possible; these are the most accurate and useful bounds. Different values of λ will give different values of $z_{\text{LR}}(\lambda)$, and hence different bounds. We'd like to find λ that gives the largest possible bounds. That is, we want to solve

$$\text{(LR)} \quad \max_{\lambda} \; z_{\text{LR}}(\lambda). \tag{D.8}$$

Suppose for now that we have found the λ^* that solves (LR). (We'll discuss one way to find such λ in Section D.1.3.) Let $z_{\text{LR}} = z_{\text{LR}}(\lambda^*)$. How good a bound is z_{LR}? For example, is it better or worse than the bound obtained from the LP relaxation of (P)? The answer turns out to be, "at least as good":

Theorem D.2

$$z_{\text{LP}} \le z_{\text{LR}},$$

where z_{LP} is the optimal objective value of the LP relaxation of (P) and z_{LR} is the optimal objective value of (LR).

Proof.

$$\begin{aligned}
z_{\text{LR}} &= \max_{\lambda} \left\{ \min_{x} \; cx + \lambda(b - Ax) \middle| \; Dx \le e, x \ge 0 \text{ and binary} \right\} \\
&\ge \max_{\lambda} \left\{ \min_{x} \; cx + \lambda(b - Ax) \middle| \; Dx \le e, x \ge 0 \right\} \\
&\qquad\qquad\qquad \text{(since relaxing integrality can't increase the objective)} \\
&= \max_{\lambda} \left\{ \min_{x} \; (c - \lambda A)x + \lambda b \middle| \; Dx \le e, x \ge 0 \right\} \\
&= \max_{\lambda} \left\{ \max_{\mu} \; \mu e + \lambda b \middle| \; \mu D \le c - \lambda A, \mu \le 0 \right\} \\
&\qquad\qquad\qquad\qquad\qquad \text{(taking LP dual of what's inside } \{\cdot\}) \\
&= \max_{\lambda,\mu} \{ \mu e + \lambda b | \; \mu D \le c - \lambda A, \mu \le 0 \} \\
&= \max_{\lambda,\mu} \{ \mu e + \lambda b | \; \mu D + \lambda A \le c, \mu \le 0 \} \\
&= \min_{y} \{ cy | \; Ay = b, Dy \le e, y \ge 0 \} \qquad \text{(taking LP dual of the entire problem)} \\
&= z_{\text{LP}}
\end{aligned}$$

■

An optimization problem with binary variables is said to have the *integrality property* if its LP relaxation always has optimal solutions that are binary. If the Lagrangian subproblem has the integrality property for all λ, then the bound from Lagrangian relaxation is exactly equal to the bound from LP relaxation:

Lemma D.3 *If* (P-LR$_\lambda$) *has the integrality property for all* λ*, then*

$$z_{\text{LP}} = z_{\text{LR}}.$$

Proof (sketch). The proof follows from the fact that the inequality in the proof of Theorem D.2 holds as an equality since removing the integrality restriction does not change the problem. ■

In a typical Lagrangian relaxation algorithm, we solve (P-LR$_\lambda$) for a given λ and then find a solution x that is feasible for (P). This is often done by modifying the solution to (P-LR$_\lambda$), converting it somehow from an infeasible solution to a feasible one. We then choose new multipliers λ in the hopes of improving the lower bound. Therefore, each iteration of the procedure consists of (1) solving (P-LR$_\lambda$), (2) finding an upper bound, and (3) updating the multipliers. In Section D.1.3, we discuss one common method for step (3).

To summarize what we have covered so far, at any given iteration of the Lagrangian relaxation procedure, we have

$$z_{\text{LR}}(\lambda) \leq z_{\text{LR}} \leq z^* \leq z(x) \tag{D.9}$$

$$z_{\text{LP}} \leq z_{\text{LR}} \leq z^* \leq z(x), \tag{D.10}$$

where

- $z_{\text{LR}}(\lambda)$ is the objective value of (P-LR$_\lambda$) for a particular λ (λ is a feasible solution to (LR))
- z_{LR} is the optimal objective value of (P-LR$_\lambda$)
- z_{LP} is the optimal objective value of the LP relaxation of (P)
- $z(x)$ is the objective value of (P) for a particular x (x is a feasible solution to (P))
- z^* is the optimal objective value of (P)

If (P-LR$_\lambda$) has the integrality property for all λ, then (D.9)–(D.10) reduce to

$$z_{\text{LR}}(\lambda) \leq z_{\text{LR}} = z_{\text{LP}} \leq z^* \leq z(x). \tag{D.11}$$

D.1.3 Subgradient Optimization

At the end of each iteration of the Lagrangian relaxation procedure, we want to update the Lagrange multipliers to coax the subproblem solution toward feasibility for (P). Let x be the optimal solution to the Lagrangian subproblem for a given λ. Consider a given constraint i and its multiplier λ_i. Should we make λ_i larger or smaller? The answer depends on whether, and how, the constraint is violated. We can write constraint i as

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \tag{D.12}$$

where n is the number of variables. We are trying to encourage the solution to satisfy this constraint by adding the penalty term

$$\lambda_i \left(b_i - \sum_{j=1}^{n} a_{ij} x_j \right) \tag{D.13}$$

to the objective function. If λ_i is too small, then there's no real penalty for making $\sum_{j=1}^{n} a_{ij} x_j$ small, and it's likely that the left-hand side of (D.12) will be too small. On the other hand, if λ_i is too large, there will be an incentive to make $\sum_{j=1}^n a_{ij}x_j$ large, making the term inside the parentheses in (D.13) negative and the overall penalty large and negative. (Remember that (P) is a minimization problem.) By changing λ_i, we can encourage $\sum_{j=1}^n a_{ij}x_j$ to be larger or smaller—hopefully equal to b_i—at the next iteration. So:

- If $\sum_{j=1}^n a_{ij}x_j < b_i$, then λ_i is too small; it should be increased.
- If $\sum_{j=1}^n a_{ij}x_j > b_i$, then λ_i is too large; it should be decreased.
- If $\sum_{j=1}^n a_{ij}x_j = b_i$, then λ_i is just right; it should not be changed.

Now, $z_{\mathrm{LR}}(\lambda)$ is a piecewise-linear concave function of λ. Solving problem (LR) involves maximizing this function. Since it's piecewise-linear (and therefore nondifferentiable at some points), we can't just take a derivative with respect to λ. Somehow, though, we want to move from our current value of λ to a better one, over and over, until we're near the maximum of the function.

We will use a common method for updating the Lagrange multipliers called *subgradient optimization*. (Other methods for nonlinear optimization, such as the volume algorithm and bundle methods, have also proved to be very effective for updating Lagrangian multipliers.) In subgradient optimization, each move consists of a step *size* (which is the same for all i) and a step *direction* (which is different for each i).

The step size at iteration t (denoted Δ^t) is computed as follows. Let $\mathcal{L}^t$ be the lower bound found at iteration t (i.e., the value of $z_{\mathrm{LR}}(\lambda)$ for the current value of λ) and let UB be the best upper bound found (i.e., the objective value of the best feasible solution found so far, by any method). Note that while $\mathcal{L}^t$ is the *last* lower bound found, UB is the *best* upper bound found. Then the step size Δ^t is given by

$$\Delta^t = \frac{\alpha^t(\mathrm{UB} - \mathcal{L}^t)}{\sum_{i=1}^m \left(b_i - \sum_{j=1}^n a_{ij}x_j\right)^2}. \tag{D.14}$$

α^t is a constant that is generally set to 2 at iteration 1 and divided by 2 after a given number (say 20) of consecutive iterations have passed during which the best known lower bound has not improved. The numerator is proportional to the difference between the upper and lower bounds—as we get closer to the maximum of the function, the steps should get smaller. The denominator is simply the sum of the squares of the constraint violations.

The step direction for constraint i is simply given by $b_i - \sum_{j=1}^n a_{ij}x_j$ (the violation in the constraint).

To obtain the new multipliers (call them λ^{t+1}) from the old ones (λ^t), we set

$$\lambda_i^{t+1} = \lambda_i^t + \Delta^t \left(b_i - \sum_{j=1}^n a_{ij}x_j\right). \tag{D.15}$$

Note that since $\Delta^t > 0$, this update step follows the rules given above:

- If $\sum_{j=1}^n a_{ij}x_j < b_i$, then λ_i increases.
- If $\sum_{j=1}^n a_{ij}x_j > b_i$, then λ_i decreases.
- If $\sum_{j=1}^n a_{ij}x_j = b_i$, then λ_i stays the same.

At the first iteration, λ can be initialized using a variety of ways: For example, set $\lambda_i = 0$ for all i, set it to some random number, or set it according to some other ad hoc rule.

D.1.4 Stopping Criteria

The process of solving (P-LR$_\lambda$), finding a feasible solution, and updating λ is continued until some stopping criteria are met. For example, we might stop the procedure when any of the following is true:

- The upper bound and lower bound are within some prespecified tolerance, say 0.1%.
- A certain number of iterations, say 1000, have elapsed.
- α^t is smaller than some pre-specified tolerance, say 10^{-6}.
- A certain amount of time, say 1 minute, has elapsed.

D.1.5 Other Problem Types

Lagrangian relaxation is a general tool that can be used for any IP. However, some of the rules discussed above change when applied to IPs that have a form other than that given in (D.1)–(D.4).

D.1.5.1 Inequality Constraints The constraints relaxed may be inequality or equality constraints.

- For $\leq$ constraints, λ is restricted to be *nonpositive*.
- For $\geq$ constraints, λ is restricted to be *nonnegative*.
- For $=$ constraints, λ is unrestricted in sign.

(*Note*: These rules assume the penalty in the objective function is written as

$$\lambda(\text{RHS} - \text{LHS}).$$

If, instead, the right-hand side is subtracted from the left-hand side, these rules are reversed.)

D.1.5.2 Maximization Problems If the IP is a maximization problem, then

- The Lagrangian subproblem provides an *upper bound* on the optimal objective value and a feasible solution provides a *lower bound*, so the relationships in (D.9) and (D.10) are reversed:

$$z(x) \leq z^* \leq z_{\text{LR}} \leq z_{\text{LR}}(\lambda) \tag{D.16}$$
$$z(x) \leq z^* \leq z_{\text{LR}} \leq z_{\text{LP}}. \tag{D.17}$$

- Problem (LR) is of the form

$$\min_{\lambda} \left\{ \begin{array}{ll} \max_x & \cdots \\ \text{s.t.} & \cdots \end{array} \right\}$$

- The $+$ sign in (D.15) becomes a $-$ sign.
- The rules for inequality constraints given in Section D.1.5.1 are reversed.

D.1.6 Branch-and-Bound

If the Lagrangian procedure stops before the upper and lower bounds are close to each other, there is no guarantee that the solution found is near-optimal. If this happens, we could stop and accept the best feasible solution found without a guarantee of optimality (this treats Lagrangian relaxation as a heuristic), or we could close the optimality gap using branch-and-bound. The branch-and-bound process is like the standard process for solving LPs except that (a) lower bounds are obtained by solving the Lagrangian subproblem, not the LP relaxation, and (b) upper bounds are found using the upper-bounding method that is embedded into the Lagrangian procedure, instead of when LP solutions happen to be integer-feasible. At each node of the branch-and-bound tree, a variable is chosen for branching, and that variable is fixed first to 0, then to 1. The mechanics of branching and fathoming are just like those in standard branch-and-bound.

D.1.7 Algorithm Summary

The Lagrangian relaxation algorithm is summarized in Algorithm D.1.

D.2 COLUMN GENERATION

D.2.1 Overview

Column generation is a useful technique for solving optimization problems, especially those in which the number of variables is much larger than the number of constraints. The number of variables may even be exponentially large—too large to enumerate all of the variables and their coefficients. The basic idea behind column generation is to optimize the problem using only a subset of the variables (the columns), and to generate new columns as needed during the algorithm. Because the vast majority of the columns will be nonbasic, i.e., will equal zero, in the optimal solution, the idea makes sense, but we must be smart, and efficient, at determining the new columns that must be generated.

Column generation was first developed for linear programming (LP) problems (Ford and Fulkerson 1958, Dantzig and Wolfe 1960), but it has since become a popular tool for solving integer programming (IP) problems, as well. It is indispensable for certain classes of IPs, such as airline crew scheduling, that were previously considered intractable.

The column generation process works as follows. The problem being solved is decomposed into two problems, called the *master problem* and the *pricing problem*. The master problem is the original problem, but we usually work with a version of it that contains only a subset of the original variables and is therefore called the *restricted master problem*. The pricing problem, also called the *subproblem*, uses dual information from the master problem to identify a column to be generated and added to the master problem. The master problem is solved again, new dual information is obtained, the pricing problem identifies a new column, and so on, until the pricing problem cannot identify a new column to add, in which case the current solution to the master problem is optimal.

Algorithm D.1 Lagrangian relaxation

input initial multipliers λ^1, initial constant α^0, α-halving constant γ, optimality tolerance κ, iteration limit ζ

$t \leftarrow 1$, LB $\leftarrow -\infty$, UB $\leftarrow \infty$, `NonImprCtr` $\leftarrow 0$ ▷ Initialization

repeat ▷ Main loop

    solve (P-LR$_\lambda$) with input λ^t to obtain $z_{\text{LR}}(\lambda^t)$ ▷ Lower bound

    if $z_{\text{LR}}(\lambda^t) > \text{LB}$ **then** ▷ Compare to best-known lower bound

        LB $\leftarrow z_{\text{LR}}(\lambda^t)$

        `NonImprCtr` $\leftarrow 0$ ▷ Reset non-improvement counter

    else

        `NonImprCtr` $\leftarrow$ `NonImprCtr` $+ 1$ ▷ Increment non-impr. counter

        if `NonImprCtr` $= \gamma$ **then** ▷ Check whether to halve α

            $\alpha^t \leftarrow \alpha^{t-1}/2$

            `NonImprCtr` $\leftarrow 0$

        else

            $\alpha^t \leftarrow \alpha^{t-1}$

        end if

    end if

    get feasible solution x for (P), with cost $z(x)$, using the solution to (P-LR$_\lambda$)

        (or some other method) ▷ Upper bound

    if $z(x) < \text{UB}$ **then** ▷ Compare to best-known upper bound

        UB $\leftarrow z(x)$

        $x^{\text{UB}} \leftarrow x$

    end if

    $\Delta^t \leftarrow \alpha^t(\text{UB} - z_{\text{LR}}(\lambda^t)) / \sum_{i=1}^m \left(b_i - \sum_{j=1}^n a_{ij}x_j\right)^2$ ▷ Update multipliers

    for all $i \in I$ **do**

        $\lambda_i^{t+1} \leftarrow \lambda_i^t + \Delta^t \left(b_i - \sum_{j=1}^n a_{ij}x_j\right)$

    end for

    $t \leftarrow t + 1$ ▷ Increment t

until $\text{UB} - z_{\text{LR}}(\lambda^t) \leq \kappa$ or $t > \zeta$ ▷ Check for termination

return x^{UB}, UB

D.2.2 Master Problem and Subproblem

Suppose we wish to solve the following LP, which we will refer to as the *master problem*:

$$
\begin{aligned}
\text{(MP)} \quad \text{minimize} \quad & \sum_{j\in J} c_j x_j && \text{(D.18)}\\
\text{subject to} \quad & \sum_{j\in J} a_j x_j \geq b && \text{(D.19)}\\
& x_j \geq 0 \qquad \forall j \in J, && \text{(D.20)}
\end{aligned}
$$

where J is the set of decision variables, c_j is a (scalar) objective function coefficient, and a_j and b are vectors of constraint coefficients. Let $\pi \geq 0$ be the vector of dual variables associated with the constraints (D.19).

Recall that a standard decision rule in the simplex method is to find the variable with the most negative reduced cost, i.e., to solve

$$c^* = \min_{j\in J} \left\{ c_j - \pi^T a_j \right\}, \tag{D.21}$$

where the superscript T stands for transpose. The $j \in J$ that minimizes (D.21) is selected as the next variable to enter the basis. If $c^* \geq 0$, i.e., $c_j - \pi^T a_j \geq 0$ for all $j \in J$, then the current solution to the LP is optimal.

Because we have assumed that the number of columns is very large, it may not be practical to solve (D.21) explicitly. Therefore, we will work with the restricted master problem, which considers only a subset $J' \subseteq J$ of the columns:

$$
\begin{aligned}
(\overline{\text{MP}}) \quad \text{minimize} \quad & \sum_{j\in J'} c_j x_j && \text{(D.22)}\\
\text{subject to} \quad & \sum_{j\in J'} a_j x_j \geq b && \text{(D.23)}\\
& x_j \geq 0 \qquad \forall j \in J' && \text{(D.24)}
\end{aligned}
$$

Let $\bar{x}$ be the optimal solution to $(\overline{\text{MP}})$ and let $\bar{\pi}$ be the corresponding optimal dual solution.

Suppose we could solve the following *pricing problem*:

$$\bar{c}^* = \min_{j\in J} \left\{ c_j - \bar{\pi}^T a_j \right\}. \tag{D.25}$$

(Note that the minimization is over all of J, not just J'.) If $\bar{c}^* \geq 0$, then the current solution $\bar{x}$ is optimal not only for $(\overline{\text{MP}})$, but also for (MP), since all of the reduced costs are nonnegative. If, instead, $\bar{c}^* < 0$, then the j that attains the minimum in (D.25) will improve the objective function value of $(\overline{\text{MP}})$, and therefore we should add it to J' and re-solve $(\overline{\text{MP}})$.

It seems unrealistic to hope to be able to solve the pricing problem, since we cannot enumerate the elements of J. However, in many cases, the pricing problem can be solved by exploiting the structure of the original problem, even without enumerating J. These are the cases in which column generation is most useful. Examples of tractable pricing problems include the cutting stock problem (Section D.2.3), the vehicle routing problem (VRP) (Section 11.2.4), and the location model with risk pooling (LMRP) (Section 12.2.7).

D.2.3 An Example: The Cutting Stock Problem

We next introduce an example to illustrate how column generation works in the context of a classic operations research (OR) problem: the one-dimensional *cutting stock problem*, introduced by Gilmore and Gomory (1965). Paper manufacturers often produce very wide rolls of paper. Their customers want narrower rolls, so the problem is to determine how to cut up the wide rolls into smaller ones while minimizing the amount of waste.

Let W be the width of each roll of paper, and let K be the set of available rolls. Let m be the number of types of rolls that are required by the customers; there is a demand b_i for type i, which has width w_i, for $i = 1, \ldots, m$. We define two sets of decision variables: $y_k = 1$ if we use roll $k \in K$, 0 otherwise; and x_{ik} equals the number of times type i is cut from roll k. The objective is to minimize the total number of rolls to be cut, while satisfying all of the customers' demands.

One way to formulate the cutting stock problem is as follows:

$$
\begin{aligned}
\text{(CS)} \qquad \text{minimize} \quad & \sum_{k \in K} y_k && && \text{(D.26)}\\
\text{subject to} \quad & \sum_{k \in K} x_i^k \geq b_i && \forall i = 1, \ldots, m && \text{(D.27)}\\
& \sum_{i=1}^{m} w_i x_i^k \leq W y_k && \forall k \in K && \text{(D.28)}\\
& y_k \in \{0, 1\} && \forall k \in K && \text{(D.29)}\\
& x_i^k \in \mathbb{Z}_+ && \forall k \in K, i = 1, ..., m && \text{(D.30)}
\end{aligned}
$$

The objective function (D.26) counts the total number of rolls used. Constraints (D.27) require the total demand of each roll type to be satisfied. Constraints (D.28) ensure that the total width of the rolls cut from roll k does not exceed W, and that no rolls are cut from k if $y_k = 0$. Constraints (D.29) and (D.30) are integrality and non-negativity constraints.

The formulation above, given by Kantorovich (1960), is natural and straightforward. However, it is not practical, because its LP relaxation is very weak. Even for moderately sized instances, it may take a very long time for an off-the-shelf IP solver to solve (D.26)–(D.30). Therefore, Gilmore and Gomory (1965) propose an alternative formulation that treats the cutting stock problem as a set covering problem.

Define a cutting pattern as a collection of roll types and quantities that can be cut from a single large roll. Let J denote the set of all feasible cutting patterns, and let a_{ij} be a parameter that equals the number of times that roll type i is cut in pattern j. A cutting pattern is feasible if it satisfies

$$\sum_{i=1}^{m} w_i a_{ij} \leq W \qquad \text{(D.31)}$$

$$a_{ij} \in \mathbb{Z}_+ \qquad \forall i = 1, ..., m. \qquad \text{(D.32)}$$

For example, suppose the width of the original roll is $W = 100$, and there are two roll types ($i = 1, 2$), with demands $b_i = 10, 20$ and widths $w_i = 25, 15$. Pattern 1 might entail the original roll being cut into three rolls of width w_1 and one roll of width w_2, in which case $a_{11} = 3$ and $a_{21} = 1$. Similarly, the large roll can also be cut into pattern 2, which might consist of two rolls of width w_1 and three of width w_2, implying that $a_{12} = 2$ and $a_{22} = 3$.

Thus, each cutting pattern j is represented by a column $(a_{1j}, a_{2j}, ..., a_{mj})^T$. The problem, of course, is that there is an exponential number of cutting patterns—but let's ignore that concern for a moment.

In our new formulation, let x_j denote the number of times that cutting pattern $j \in J$ is used, i.e., the number of large rolls that will be cut using pattern j. Then we can reformulate the cutting stock problem as follows:

$$\begin{aligned}
\text{(CSE)} \qquad & \text{minimize} \quad && \sum_{j \in J} x_j && && \text{(D.33)} \\
& \text{subject to} \quad && \sum_{j \in J} a_{ij} x_j \ge b_i && \forall i = 1, \ldots, m && \text{(D.34)} \\
& && x_j \in \mathbb{Z}_+ && \forall j \in J && \text{(D.35)}
\end{aligned}$$

This formulation is sometimes called the *extensive* formulation, since it contains many more decision variables, whereas (CS) is called the *compact* formulation.

The idea is to solve the LP relaxation of (CSE) using column generation. The LP relaxation will be our master problem. It turns out that the pricing problem is relatively easy—it is a knapsack problem. Of course, the LP relaxation may not provide a feasible integer solution, but it might provide a tight lower bound for the original problem, because the extensive formulation has a tighter LP bound than the compact formulation does. The LP relaxation can also be used to develop heuristic solutions to the integer problem.

As we noted above, J is exponentially large. Therefore, we will work with the restricted master problem, in which the LP relaxation uses only the patterns in $J' \subseteq J$, which is typically much smaller than J. The restricted master problem is given by:

$$\begin{aligned}
(\overline{\text{CSE}}) \qquad & \text{minimize} \quad && \sum_{j \in J'} x_j && && \text{(D.36)} \\
& \text{subject to} \quad && \sum_{j \in J'} a_{ij} x_j \ge b_i && \forall i = 1, \ldots, m && \text{(D.37)} \\
& && x_j \ge 0 && \forall j \in J' && \text{(D.38)}
\end{aligned}$$

The initial subset J' can be calculated using a simple approach. For example, we might generate a column for each roll type i, with $\lfloor W/w_i \rfloor$ rolls of type i cut from the original roll, i.e., $a_{ii} = \lfloor W/w_i \rfloor$.

We can solve $(\overline{\text{CSE}})$ using standard LP algorithms. Let $\bar{\pi}$ be the vector of optimal dual variables. Then the reduced cost of a primal variable x_j is

$$1 - \sum_{i=1}^{m} a_{ij} \bar{\pi}_i.$$

Our task, then, is to identify a column (a cutting pattern) in $J \setminus J'$ that would improve the objective function of the restricted master problem $(\overline{\text{CSE}})$ if we were to add it to J'. Such a column would have a negative reduced cost. Thus, we would like to solve a problem like (D.25), but we need to do it without enumerating all of J.

Recall that a cutting pattern is feasible if it satisfies (D.31)–(D.32). We can formulate an optimization problem for "pricing out" the desired column, i.e., the one with the most negative reduced cost. This pricing problem is given by:

$$\text{minimize} \quad 1 - \sum_{i=1}^{m} \bar{\pi}_i a_i \tag{D.39}$$

$$\text{subject to} \quad \sum_{i=1}^{m} w_i a_i \leq W \tag{D.40}$$

$$a_i \in \mathbb{Z}_+ \qquad \forall i = 1, \ldots, m, \tag{D.41}$$

where we replace a_{ij} with a_i for convenience, and since we are only concerned with finding one j. Fortunately, (D.39)–(D.41) is easy to solve—it is simply an integer knapsack problem. (Think of multiplying the objective function by -1 to convert it to a maximization problem and omitting the constant 1.) Note that the a_i are now the decision variables: We are trying to find a cutting pattern, as defined by the a_i, that is feasible and that minimizes the reduced cost. If the optimal objective function value of (D.39) is negative, then we have found a new column, defined by the a_i, to add to the restricted master problem. On the other hand, if the optimal objective of the pricing problem is nonnegative, then we know there is no cutting pattern that has negative reduced cost and that will improve the restricted master problem. We can conclude that we have found the optimal solution to the LP relaxation of the (full) master problem.

Moreover, the LP solution can be converted to an integer solution by rounding the fractional x_j up. Since (D.34) holds for the fractional solution, it also holds for the rounded-up solution.

D.2.4 Column Generation for Integer Programs

Column generation was originally designed for solving linear programs. However, it can also be used to solve large-scale integer programs together with the classical branch-and-bound framework. This method is known as *branch-and-price*. In particular, the original integer program is converted to an extensive formulation using an approach known as Dantzig–Wolfe decomposition. The extensive formulation typically has tighter LP relaxations and often eliminates symmetry. In the branch-and-price algorithm, we use column generation to solve the LP relaxation at each node of the branch-and-bound tree. Other aspects are similar to classical branch-and-bound, although the branching strategy is often somewhat different. A generic procedure for the branch-and-price algorithm is introduced by Barnhart et al. (1998).

Other practical issues arise when implementing column generation for integer programs. For example, one must consider how to reformulate the original problem into an appropriate extensive form so that both the restricted master problem and the pricing subproblem are tractable. Moreover, the solution process may exhibit a so-called *tailing-off effect*, in which the convergence becomes significantly slower after a near-optimal solution is found. For further discussion on column generation for integer programming, we refer the readers to tutorials such as Wilhelm (2001) and Lübbecke and Desrosiers (2005).

References

Y. Adulyasak, J.-F. Cordeau, and R. Jans. Formulations and branch-and-cut algorithms for multivehicle production and inventory routing problems. *INFORMS Journal on Computing*, 26(1): 103–120, 2014.

Y. Agarwal, K. Mathur, and H. M. Salkin. A set-partitioning-based exact algorithm for the vehicle routing problem. *Networks*, 19(7):731–749, 1989.

R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows*. Prentice Hall, Upper Saddle River, NJ, 1993.

A. A. Akhil, G. Huff, A. B. Currier, B. C. Kaun, D. M. Rastler, S. B. Chen, A. L. Cotter, D. T. Bradshaw, and W. D. Gauntlett. DOE/EPRI 2013 electricity storage handbook in collaboration with NRECA. Report SAND2013-5131, Sandia National Laboratories, July 2013.

U. Akinc and B. M. Khumawala. An efficient branch and bound algorithm for the capacitated warehouse location problem. *Management Science*, 23(6):585–594, 1977.

E. Aktaş, O. Özaydın, B. Bozkaya, F. Ülengin, and Ş. Önsel. Optimizing fire station locations for the Istanbul Metropolitan Municipality. *Interfaces*, 43(3):240–255, 2013.

K. Al-Sultan and M. Al-Fawzan. A tabu search approach to the uncapacitated facility location problem. *Annals of Operations Research*, 86:91–103, 1999.

N. Alguacil, A. L. Motto, and A. J. Conejo. Transmission expansion planning: A mixed-integer LP approach. *IEEE Transactions on Power Systems*, 18(3):1070–1077, 2003.

Ö. G. Ali, S. Sayın, T. Van Woensel, and J. Fransoo. SKU demand forecasting in the presence of promotions. *Expert Systems with Applications*, 36(10):12340–12348, 2009.

N. Altay and W. G. Green. OR/MS research in disaster operations management. *European Journal of Operational Research*, 175(1):475–493, 2006.

S. Alumur and B. Y. Kara. Network hub location problems: The state of the art. *European Journal of Operational Research*, 190(1):1–21, 2008.

Fundamentals of Supply Chain Theory, Second Edition. Lawrence V. Snyder and Zuo-Jun Max Shen.

Companion website: www.wiley.com/go/Snyder/SupplyChainTheory

R. M. Anderson and R. M. May. *Infectious Diseases of Humans: Dynamics and Control*. Oxford University Press, New York, 1992.

D. Applegate, R. Bixby, V. Chvátal, and B. Cook. Finding cuts in the TSP (a preliminary report). Technical report, DIMACS, Rutgers University, New Brunswick, NJ, 1995.

D. Applegate, R. Bixby, V. Chvátal, and W. J. Cook. Concorde (software), 2006. URL `http://www.math.uwaterloo.ca/tsp/concorde`.

D. L. Applegate, V. Chvátal, W. J. Cook, and R. E. Bixby. *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, Princeton, NJ, 2007.

C. Archetti, L. Bertazzi, G. Laporte, and M. G. Speranza. A branch-and-cut algorithm for a vendor-managed inventory–routing problem. *Transportation Science*, 41(3):382–391, 2007.

B. C. Archibald and E. A. Silver. (s, S) policies under continuous review and discrete compound poisson demand. *Management Science*, 24(9):899–909, 1978.

S. Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *Journal of the ACM*, 45(5):753–782, 1998.

S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998.

M. A. Arostegui, Jr., S. N. Kadipasaoglu, and B. M. Khumawala. An empirical comparison of tabu search, simulated annealing, and genetic algorithms for facilities location problems. *International Journal of Production Economics*, 103(2):742–754, 2006.

K. J. Arrow, T. Harris, and J. Marschak. Optimal inventory policy. *Econometrica*, 19(3):250–272, 1951.

A. Atamtürk. On capacitated network design cut–set polyhedra. *Mathematical Programming*, 92(3): 425–437, 2002.

A. Atamturk, G. Berenguer, and Z.-J. M. Shen. A conic integer programming approach to stochastic joint location-inventory problems. *Operations Research*, 60(2):366–381, 2012.

L. M. Ausubel and P. Milgrom. The lovely but lonely Vickrey auction. In P. Cramton, Y. Shoham, and R. Steinberg, editors, *Combinatorial Auctions*, chapter 1, pages 17–40. MIT Press, 2006.

I. Averbakh and O. Berman. Algorithms for the robust 1-center problem on a tree. *European Journal of Operational Research*, 123(2):292–302, 2000a.

I. Averbakh and O. Berman. Minmax regret median location on a network under uncertainty. *INFORMS Journal on Computing*, 12(2):104–110, 2000b.

A. Avrahami, Y. T. Herer, and R. Levi. Matching supply and demand: Delayed two-phase distribution at Yedioth Group—models, algorithms, and information technology. *Interfaces*, 44(5):445–460, 2014.

S. Axsäter. Simple solution procedures for a class of two-echelon inventory problems. *Operations Research*, 38(1):64–69, 1990.

S. Axsäter. Using the deterministic EOQ formula in stochastic inventory control. *Management Science*, 42(6):830–834, 1996.

B. Aytac and S. D. Wu. Characterization of demand for short life-cycle technology products. *Annals of Operations Research*, 203(1):255–277, 2013.

E. Babiloni, E. Guijarro, M. Cardós, and S. Estellés. Exact fill rates for the (R, S) inventory control with discrete distributed demands for the backordering case. *Informatica Economică*, 16(3): 19–26, 2012.

G. Babin, S. Deneault, and G. Laporte. Improvements to the Or-opt heuristic for the symmetric travelling salesman problem. *Journal of the Operational Research Society*, 58:402–407, 2007.

F. Babonneau, Y. Nesterov, and J.-P. Vial. Design and operations of gas transmission networks. *Operations Research*, 60(1):34–47, 2012.

A. Balakrishnan, T. L. Magnanti, and R. T. Wong. A dual-ascent procedure for large-scale uncapacitated network design. *Operations Research*, 37(5):716–740, 1989.

E. Balas and P. Toth. Branch and bound methods. In E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, editors, *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, chapter 10, pages 361–401. John Wiley & Sons, Hoboken, NJ, 1985.

B. Balcik, B. M. Beamon, and K. Smilowitz. Last mile distribution in humanitarian relief. *Journal of Intelligent Transportation Systems*, 12(2):51–63, 2008.

M. L. Balinski. Integer programming: Methods, uses, computation. *Management Science*, 12(3): 253–313, 1965.

M. L. Balinski and R. E. Quandt. On an integer program for a delivery problem. *Operations Research*, 12(2):300–304, 1964.

R. H. Ballou. Dynamic warehouse location analysis. *Journal of Marketing Research*, 5:271–276, 1968.

M. Banciu and P. Mirchandani. Technical note—new results concerning probability distributions with increasing generalized failure rates. *Operations Research*, 61(4):925–931, 2013.

F. Barahona and D. Jensen. Plant location with minimum inventory. *Mathematical Programming*, 83:101–111, 1998.

J. Barcelo, E. Fernandez, and K. O. Jörnsten. Computational results from a new Lagrangean relaxation algorithm for the capacitated plant location problem. *European Journal of Operational Research*, 53(1):38–45, 1991.

C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3): 316–329, 1998.

J. J. Bartholdi and L. K. Platzman. An $O(N \log N)$ planar travelling salesman heuristic based on spacefilling curves. *Operations Research Letters*, 1(4):121–125, 1982.

J. J. Bartholdi, L. K. Platzman, R. L. Collins, and W. H. Warden. A minimal technology routing system for Meals on Wheels. *Interfaces*, 13(3):1–8, 1983.

F. Bass. A new product growth model for consumer durables. *Management Science*, 15(5):215–227, 1969.

F. Bass. Comments on "A new product growth for model consumer durables": The Bass model. *Management Science*, 50(12S):1833–1840, 2004.

J. Beardwood, J. H. Halton, and J. M. Hammersley. The shortest path through many points. *Mathematical Proceedings of the Cambridge Philosophical Society*, 55(4):299–327, 1959.

J. E. Beasley. Route first–cluster second methods for vehicle routing. *OMEGA*, 11(4):403–408, 1983.

S. Beggs, S. Cardell, and J. Hausman. Assessing the potential demand for electric cars. *Journal of Econometrics*, 16:1–19, 1981.

C. E. Bell. Improved algorithms for inventory and replacement-stocking problems. *SIAM Journal on Applied Mathematics*, 18(3):558–566, 1970.

R. Bellman. Dynamic programming treatment of the travelling salesman problem. *Journal of the ACM*, 9(1):61–63, 1962.

E. J. Beltrami and L. D. Bodin. Networks and vehicle routing for municipal waste collection. *Networks*, 4(1):65–94, 1974.

M. Ben-Akiva and S. Lerman. *Discrete Choice Analysis: Theory and Application to Travel Demand*. MIT Press, Cambridge, MA, 1985.

G. Benoit and S. Boyd. Finding the exact integrality gap for small traveling salesman problems. *Mathematics of Operations Research*, 33(4):921–931, 2008.

J. J. Bentley. Fast algorithms for geometric traveling salesman problems. *ORSA Journal on Computing*, 4(4):387–411, 1992.

R. T. Berger, C. R. Coullard, and M. S. Daskin. Location-routing problems with distance constraints. *Transportation Science*, 41(1):29–43, 2007.

E. Berk and A. Arreola-Risa. Note on "Future supply uncertainty in EOQ models". *Naval Research Logistics*, 41(1):129–132, 1994.

O. Berman, D. Krass, and M. B. C. Menezes. Facility reliability issues in network p-median problems: Strategic centralization and co-location effects. *Operations Research*, 55(2):332–350, 2007.

D. Bienstock and O. Günlük. Capacitated network design—Polyhedral structure and computation. *INFORMS Journal on Computing*, 8(3):243–259, 1996.

M. Bijvank and I. F. Vis. Lost-sales inventory theory: A review. *European Journal of Operational Research*, 215(1):1–13, 2011.

S. Biller, A. Muriel, and Y. Zhang. Impact of price postponement on capacity and flexibility investment decisions. *Production and Operations Management*, 15(2):198–214, 2006.

S. Binato, M. V. F. Pereira, and S. Granville. A new Benders decomposition approach to solve power transmission network design problems. *IEEE Transactions on Power Systems*, 16(2):235–240, 2001.

A. Bixby. *Polyhedral Analysis and Effective Algorithms for the Capacitated Vehicle Routing Problem*. Ph.D. dissertation, Northwestern University, Evanston, IL, 1998.

C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3):268–308, 2003.

N. Boudette. Chrysler gains edge by giving new flexibility to its factories. *Wall Street Journal*, April 11, 2006.

S. Boyd and R. Carr. A new bound for the ratio between the 2-matching problem and its linear programming relaxation. *Mathematical Programming Series A*, 86:499–514, 1999.

S. Boyd and R. Carr. Finding low cost TSP and 2-matching solutions using certain half-integer subtour vertices. *Discrete Optimization*, 8(4):525–539, 2011.

S. Boyd and P. Elliott-Magwood. The structure of the extreme points of the subtour elimination polytope of the STSP. In S. Iwata, editor, *Combinatorial Optimization and Discrete Algorithms*, volume B23 of *RIMS Kôkyûroku Bessatsu*. Research Institute for Mathematical Sciences, Kyoto University, Kyoto, Japan, 2010.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, 7th edition, 2009.

S. C. Boyd, W. H. Cunningham, M. Queyranne, and Y. Wang. Ladders for travelling salesmen. *SIAM Journal on Optimization*, 5:408–420, 1995.

J. Bramel and D. Simchi-Levi. A location based heuristic for general routing problems. *Operations Research*, 43(4):649–660, 1995.

J. Bramel and D. Simchi-Levi. Probabilistic analysis and practical algorithms for the vehicle routing problem with time windows. *Operations Research*, 44:501–509, 1996.

J. Bramel and D. Simchi-Levi. On the effectiveness of set covering formulations for the vehicle routing problem with time windows. *Operations Research*, 45(2):295–301, 1997.

J. Bramel and D. Simchi-Levi. Set-covering-based algorithms for the capacitated VRP. In P. Toth and D. Vigo, editors, *The Vehicle Routing Problem*, chapter 4, pages 85–108. Society for Industrial and Applied Mathematics, Philadelphia, 2001.

J. Bramel, E. G. Coffman, P. W. Shor, and D. Simchi-Levi. Probabilistic analysis of the capacitated vehicle routing problem with unsplit demands. *Operations Research*, 40(6):1095–1106, 1992.

P. D. Brown, J. A. P. Lopes, and M. A. Matos. Optimization of pumped storage capacity in an isolated power system with large renewable penetration. *IEEE Transactions on Power Systems*, 23(2): 523–531, 2008.

S. Browne and P. Zipkin. Inventory models with continuous, stochastic demands. *Annals of Applied Probability*, 1(3):419–435, 1991.

C. A. Butler and J. D. Camm. Introduction: The 2009 Daniel H. Wagner Prize for excellence in operations research practice. *Interfaces*, 40(5):339–341, 2010.

G. P. Cachon. Supply chain coordination with contracts. In A. G. de Kok and S. C. Graves, editors, *Supply Chain Management: Design, Coordination and Operation*, volume 11 of *Handbooks in Operations Research and Management Science*, chapter 6. Elsevier, New York, 2003.

G. P. Cachon and M. A. Lariviere. Supply chain coordination with revenue-sharing contracts: Strengths and limitations. *Management Science*, 51(1):30–44, 2005.

G. P. Cachon and S. Netessine. Game theory in supply chain analysis. In D. Simchi-Levi, S. D. Wu, and Z.-J. M. Shen, editors, *Handbook of Quantitative Supply Chain Analysis: Modeling in the E-Business Era*, chapter 2. Springer, New York, 2004.

P. Camerini, L. Fratta, and F. Maffioli. On improving relaxation methods by modified gradient techniques. In M. L. Balinski and P. Wolfe, editors, *Nondifferentiable Optimization*, volume 3 of *Mathematical Programming Studies*, pages 26–34. Springer, Berlin, 1975.

J. F. Campbell. Integer programming formulations of discrete hub location problems. *European Journal of Operational Research*, 72(2):387–405, 1994a.

J. F. Campbell. A survey of network hub location. *Studies in Locational Analysis*, 6(6):31–49, 1994b.

J. F. Campbell. Hub location and the p-hub median problem. *Operations Research*, 44(6):923–935, 1996.

R. Carbonneau, K. Laframboise, and R. Vahidov. Application of machine learning techniques for supply chain demand forecasting. *European Journal of Operational Research*, 184(3):1140–1154, 2008.

B. Carlson, Y. Chen, M. Hong, R. Jones, K. Larson, X. Ma, P. Nieuwesteeg, H. Song, K. Sperry, M. Tackett, D. Taylor, J. Wan, and E. Zak. MISO unlocks billions in savings through the application of operations research for energy and ancillary services markets. *Interfaces*, 42(1): 58–73, 2012.

E. D. Castronuovo and J. A. P. Lopes. On the optimization of the daily operation of a wind-hydro power plant. *IEEE Transactions on Power Systems*, 19(3):1599–1606, 2004.

A. M. Caunhye, X. Nie, and S. Pokharel. Optimization models in emergency logistics: A literature review. *Socio-Economic Planning Sciences*, 46(1):4–13, 2012.

M. Çelik, Ö. Ergun, B. Johnson, P. Keskinocak, Á. Lorca, P. Pekgün, and J. Swann. Humanitarian logistics. In P. B. Mirchandani, editor, *INFORMS Tutorials in Operations Research*, chapter 2, pages 18–49. INFORMS, Hanover, MD, 2012.

M. Çelik, Ö. Ergun, P. Keskinocak, M. Soldner, and J. Swann. Humanitarian applications of supply chain optimization. In T. Terlaky, M. Anjos, and S. Ahmed, editors, *Advances and Trends in Optimization with Engineering Applications*, chapter 36, pages 479–492. SIAM, Philadelphia, 2017.

S. Çetinkaya, H. Üster, G. Easwaran, and B. B. Keskin. An integrated outbound logistics model for Frito-Lay: Coordinating aggregate-level production and distribution decisions. *Interfaces*, 39 (5):460–475, 2009.

L. M. A. Chan, A. Muriel, Z.-J. M. Shen, and D. Simchi-Levi. On the effectiveness of zero-inventory-ordering policies for the economic lot-sizing model with a class of piecewise linear cost structures. *Operations Research*, 50(6):1058–1067, 2002.

R. Chandrasekaran and A. Daughety. Location on tree networks: p-centre and n-dispersion problems. *Mathematics of Operations Research*, 6(1):50–57, 1981.

B. Chen and C.-S. Lin. Minmax-regret robust 1-median location on a tree. *Networks*, 31(2):93–103, 1998.

F. Chen and R. Samroengraja. The stationary beer game. *Production and Operations Management*, 9(1):19–30, 2000.

F. Chen and Y.-S. Zheng. Lower bounds for multi-echelon stochastic inventory systems. *Management Science*, 40(11):1426–1443, 1994.

F. Chen, Z. Drezner, J. K. Ryan, and D. Simchi-Levi. Quantifying the bullwhip effect in a simple supply chain: The impact of forecasting, lead times, and information. *Management Science*, 46(3):436–443, 2000.

R. R. Chen, R. O. Roundy, R. Q. Zhang, and G. Janakiraman. Efficient auction mechanisms for supply chain procurement. *Management Science*, 51(3):467–482, 2005.

W. Chen, M. Dawande, and G. Janakiraman. Fixed-dimensional stochastic dynamic programs: An approximation scheme and an inventory application. *Operations Research*, 62(1):81–103, 2014.

R. Cheung and W. B. Powell. Models and algorithms for distribution problems with uncertain demands. *Transportation Science*, 30(1):43–59, 1996.

Chicago Data Portal. Libraries—locations, hours and contact information [data set], June 2, 2017a. URL `https://data.cityofchicago.org/Education/Libraries-Locations-Hours-and-Contact-Information/x8fc-8rcq`.

Chicago Data Portal. Chicago public schools—school locations SY1718 [data set], September 5, 2017b. URL `https://data.cityofchicago.org/Education/Chicago-Public-Schools-School-Locations-SY1718/4g38-vs8v`.

S. E. Chick, H. Mamani, and D. Simchi-Levi. Supply chain coordination and influenza vaccination. *Operations Research*, 56(6):1493–1506, 2008.

F. Chiyoshi and R. D. Galvão. A statistical analysis of simulated annealing applied to the p-median problem. *Annals of Operations Research*, 96(1):61–74, 2000.

S. Chopra and P. Meindl. *Supply Chain Management: Strategy, Planning and Operation*. Prentice-Hall, Upper Saddle River, NJ, 2001.

M. C. Chou, C.-P. Teo, and H. Zheng. Process flexibility: Design, evaluation and applications. *Flexible Services and Manufacturing Journal*, 20:59–94, 2008.

M. C. Chou, G. A. Chua, and C.-P. Teo. On range and response: Dimensions of process flexibility. *European Journal of Operational Research*, 207(2):711–724, 2010a.

M. C. Chou, G. A. Chua, C.-P. Teo, and H. Zheng. Design for process flexibility: Efficiency of the long chain and sparse structure. *Operations Research*, 58(1):43–58, 2010b.

M. C. Chou, G. A. Chua, C.-P. Teo, and H. Zheng. Process flexibility revisited: The graph expander and its applications. *Operations Research*, 59(5):1090–1105, 2011.

N. Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Management Sciences Research Report 388, Carnegie-Mellon University, 1976.

N. Christofides and J. E. Beasley. Extensions to a Lagrangean relaxation approach for the capacitated warehouse location problem. *European Journal of Operational Research*, 12(1):19–28, 1983.

N. Christofides and J. E. Beasley. The period routing problem. *Networks*, 14(2):237–256, 1984.

N. Christofides and S. Eilon. An algorithm for the vehicle-dispatching problem. *Operational Research Quarterly*, 20(3):309–318, 1969.

N. Christofides, A. Mingozzi, and P. Toth. The vehicle routing problem. In N. Christofides, A. Mingozzi, and P. Toth, editors, *Combinatorial Optimization*, pages 315–338. Wiley, Chichester, 1979.

N. Christofides, A. Mingozzi, and P. Toth. State-space relaxation procedures for the computation of bounds to routing problems. *Networks*, 11(2):145–164, 1981.

L. Chu and Z.-J. M. Shen. Note on the complexity of the safety stock placement problem. Technical note, University of Florida, 2003.

L. Y. Chu and Z.-J. M. Shen. Agent competition double auction mechanism. *Management Science*, 52:1215–1222, 2006.

L. Y. Chu and Z.-J. M. Shen. Truthful double auction mechanisms. *Operations Research*, 56:102–120, 2008.

R. Church and C. ReVelle. The maximal covering location problem. *Papers of the Regional Science Association*, 32:101–118, 1974.

R. L. Church and R. S. Garfinkel. Locating an obnoxious facility on a network. *Transportation Science*, 12(2):107–118, 1978.

F. W. Ciarallo, R. Akella, and T. E. Morton. A periodic review, production planning model with uncertain capacity and uncertain demand—optimality of extended myopic policies. *Management Science*, 40(3):320–332, 1994.

A. J. Clark and H. Scarf. Optimal policies for a multi-echelon inventory problem. *Management Science*, 6(4):475–490, 1960.

G. Clarke and J. W. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4):568–581, 1964.

L. C. Coelho and G. Laporte. The exact solution of several classes of inventory–routing problems. *Computers & Operations Research*, 40(2):558–565, 2013.

L. C. Coelho, J.-F. Cordeau, and G. Laporte. Thirty years of inventory routing. *Transportation Science*, 48(1):1–19, 2013.

J. L. Cohon. *Multiobjective Programming and Planning*. Academic Press, New York, 1978.

B. Colson, P. Marcotte, and G. Savard. An overview of bilevel optimization. *Annals of Operations Research*, 153(1):235–256, 2007.

W. J. Cook. *In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation*. Princeton University Press, Princeton, NJ, 2012.

W. J. Cook. The traveling salesman problem. Website, 2018a. URL `http://www.math.uwaterloo.ca/tsp/index.html`.

W. J. Cook. Concorde TSP app. Software, 2018b.

W. J. Cook and V. Chvátal. The birth of the cutting-plane method. In M. Jünger, T. M. Liebling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, and L. A. Wolsey, editors, *50 Years of Integer Programming 1958–2008: From the Early Years to the State-of-the-Art*, pages 7–9. Springer, Berlin, 2010.

C. J. Corbett and K. Rajaram. A generalization of the inventory pooling effect to nonnormal dependent demand. *Manufacturing & Service Operations Management*, 8(4):351–358, 2006.

J.-F. Cordeau and G. Laporte. Tabu search heuristics for the vehicle routing problem. In R. Sharda, S. Voss, C. Rego, and B. Alidaee, editors, *Metaheuristic Optimization via Memory and Evolution*, volume 30 of *Operations Research/Computer Science Interfaces Series*, pages 145–163. Springer US, New York, 2005.

J.-F. Cordeau, M. Gendreau, and G. Laporte. A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, 30(2):105–119, 1997.

J.-F. Cordeau, G. Desaulniers, J. Desrosiers, M. M. Solomon, and F. Soumis. VRP with time windows. In P. Toth and D. Vigo, editors, *The Vehicle Routing Problem*, chapter 7, pages 157–193. Society for Industrial and Applied Mathematics, Philadelphia, 2001.

J.-F. Cordeau, M. Gendreau, G. Laporte, J.-Y. Potvin, and F. Semet. A guide to vehicle routing heuristics. *Journal of the Operational Research Society*, 53(5):512–522, 2002.

G. Cornuejols and F. Harche. Polyhedral study of the capacitated vehicle routing problem. *Mathematical Programming*, 60(1–3):21–52, 1993.

G. Cornuejols, M. L. Fisher, and G. L. Nemhauser. Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms. *Management Science*, 23(8):789–810, 1977.

G. Cornuéjols, J. Fonlupt, and D. Naddef. The traveling salesman problem on a graph and some related integer polyhedra. *Mathematical Programming Series B*, 33(1):1–27, 1985.

G. Cornuejols, R. Sridharan, and J. Thizy. A comparison of heuristics and relaxations for the capacitated plant location problem. *European Journal of Operational Research*, 50:280–297, 1991.

A. M. Costa. A survey on Benders decomposition applied to fixed-charge network design problems. *Computers & Operations Research*, 32(6):1429–1450, 2005.

Council of Supply Chain Management Professionals. Annual state of logistics report: Steep grade ahead, 2018a.

Council of Supply Chain Management Professionals. CSCMP supply chain management definitions and glossary, 2018b. URL `https://cscmp.org/CSCMP/Educate/SCM_Definitions_and_Glossary_of_Terms/CSCMP/Educate/SCM_Definitions_and_Glossary_of_Terms.aspx`.

T. G. Crainic, M. Gendreau, and J. M. Farvolden. A simplex-based tabu search method for capacitated network design. *INFORMS Journal on Computing*, 12(3):223–236, 2000.

T. G. Crainic, A. Frangioni, and B. Gendron. Bundle-based relaxation methods for multicommodity capacitated fixed charge network design. *Discrete Applied Mathematics*, 112(1):73–99, 2001.

G. A. Croes. A method for solving traveling-salesman problems. *Operations Research*, 6(6):791–812, 1958.

H. Crowder and M. W. Padberg. Solving large-scale symmetric travelling salesman problems to optimality. *Management Science*, 26(5):495–509, 1980.

T. Cui, Y. Ouyang, and Z.-J. M. Shen. Reliable facility location design under the risk of disruptions. *Operations Research*, 58(4):998–1011, 2010.

M. Dada, N. C. Petruzzi, and L. B. Schwarz. A newsvendor's procurement problem when suppliers are unreliable. *Manufacturing & Service Operations Management*, 9(1):9–32, 2007.

G. B. Dantzig and J. H. Ramser. The truck dispatching problem. *Management Science*, 6(1):80–91, 1959.

G. B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 8 (1):101–111, 1960.

G. B. Dantzig, D. R. Fulkerson, and S. Johnson. Solution of a large-scale traveling-salesman problem. *Journal of the Operations Research Society of America*, 2(4):393–410, 1954.

G. B. Dantzig, D. R. Fulkerson, and S. M. Johnson. On a linear-programming, combinatorial approach to the traveling-salesman problem. *Operations Research*, 7(1):58–66, 1959.

A. Dasci. Conditional location problems on networks and in the plane. In H. A. Eiselt and V. Marianov, editors, *Foundations of Location Analysis*, chapter 9, pages 179–206. Springer-Verlag, Berlin, 2011.

M. S. Daskin. *Network and Discrete Location: Models, Algorithms, and Applications*. Wiley, New York, 1995.

M. S. Daskin. A new approach to solving the vertex p-center problem to optimality: Algorithm and computational results. *Communications of the Operations Research Society of Japan*, 45(9): 428–436, 2000.

M. S. Daskin. *Service Science*. Wiley, Hoboken, NJ, 2010.

M. S. Daskin. *Network and Discrete Location: Models, Algorithms, and Applications*. Wiley, New York, 2nd edition, 2013.

M. S. Daskin, C. R. Coullard, and Z.-J. M. Shen. An inventory-location model: Formulation, solution algorithm and computational results. *Annals of Operations Research*, 110(1–4):83–106, 2002.

C. d'Aspremont, J. J. Gabszewicz, and J.-F. Thisse. On Hotelling's "stability in competition". *Econometrica*, 47(5):1145–1150, 1979.

T. Davenport. Analytics 3.0. *Harvard Business Review*, December 2013.

P. Davis and T. L. Ray. A branch-bound algorithm for the capacitated facilities location problem. *Naval Research Logistics Quarterly*, 16:331–344, 1969.

T. de Kok, F. Janssen, J. van Doremalen, E. van Wachem, M. Clerkx, and W. Peeters. Philips Electronics synchronizes its supply chain to end the bullwhip effect. *Interfaces*, 35(1):37–48, 2005.

T. G. de Kok and J. C. Fransoo. Planning supply chain operations: Definition and comparison of planning concepts. In *Supply Chain Management: Design, Coordination and Operation*, volume 11 of *Handbooks in Operations Research and Management Science*, pages 597–675. Elsevier, New York, 2003.

T. G. de Kok and J. W. Visschers. Analysis of assembly systems with service level constraints. *International Journal of Production Economics*, 59(1):313–326, 1999.

S. de Vries and R. V. Vohra. Combinatorial auctions: A survey. *INFORMS Journal on Computing*, 15(3):284–309, 2003.

D. De Wolf and Y. Smeers. The gas transmission problem solved by an extension of the simplex algorithm. *Management Science*, 46(11):1454–1465, 2000.

I. Deif and L. D. Bodin. Extension of the Clarke and Wright algorithm for solving the vehicle routing problem with backhauling. In A. Kidder, editor, *Proceedings of the Babson College Conference on Software Uses in Transportation and Logistic Management*, pages 75–96, Babson Park, MA, 1984.

S. DeNegre and T. K. Ralphs. A branch-and-cut algorithm for integer bilevel linear programs. In *Operations Research and Cyber-Infrastructure*, pages 65–78. Springer, New York, 2009.

G. Desaulniers, J. Desrosiers, A. Erdmann, M. M. Solomon, and F. Soumis. VRP with pickup and delivery. In P. Toth and D. Vigo, editors, *The Vehicle Routing Problem*, chapter 9, pages 225–242. Society for Industrial and Applied Mathematics, Philadelphia, 2001.

K. DesMarteau. Leading the way in changing times. *Bobbin*, 41(2):48–54, 1999.

M. Desrochers and T. W. Verhoog. A matching based savings algorithm for the vehicle routing problem. Les Cahiers du GERAD G-89-04, École des Hautes Études Commerciales de Montréal, 1989.

M. Desrochers, J. Desrosiers, and M. Solomon. A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research*, 40(2):342–354, 1992.

M. Drexl and M. Schneider. A survey of variants and extensions of the location-routing problem. *European Journal of Operational Research*, 241(2):283–308, 2015.

Z. Drezner. Competitive location strategies for two facilities. *Regional Science and Urban Economics*, 12(4):485–493, 1982.

Z. Drezner, editor. *Facility Location: A Survey of Applications and Methods*. Springer-Verlag, New York, 1995a.

Z. Drezner. Dynamic facility location: The progressive p-median problem. *Location Science*, 3(1): 1–7, 1995b.

Z. Drezner and H. W. Hamacher, editors. *Facility Location: Applications and Theory*. Springer-Verlag, New York, 2002.

Z. Drezner and S. Salhi. Using hybrid metaheuristics for the one-way and two-way network design problem. *Naval Research Logistics*, 49(5):449–463, 2002.

Z. Drezner and G. O. Wesolowsky. Facility location when demand is time dependent. *Naval Research Logistics*, 38(5):763–777, 1991.

L. M. Drummond, L. S. Ochi, and D. S. Vianna. An asynchronous parallel metaheuristic for the period vehicle routing problem. *Future Generation Computer Systems*, 17(4):379–386, 2001.

Y. P. Dusonchet and A. El-Abiad. Transmission planning using discrete dynamic optimizing. *IEEE Transactions on Power Apparatus and Systems*, PAS-92(4):1358–1371, 1973.

F. Edgeworth. The mathematical theory of banking. *Journal of the Royal Statistical Society*, 51: 113–127, 1888.

J. Edmonds. Maximum matching and a polyhedron with 0,1-vertices. *Journal of Research of the National Bureau of Standards–B*, 69B(1 and 2):125–130, 1965.

M. A. Efroymson and T. L. Ray. A branch-bound algorithm for plant location. *Operations Research*, 14(3):361–368, 1966.

R. Ehrhardt. The power approximation for computing (s, S) inventory policies. *Management Science*, 25(8):777–786, 1979.

R. Ehrhardt and C. Mosier. A revision of the power approximation for computing (s, S) policies. *Management Science*, 30(5):618–622, 1984.

S. Eilon, C. D. T. Watson-Gandy, and N. Christofides. *Distribution Management: Mathematical Modelling and Practical Analysis*. Griffin, London, 1971.

H. A. Eiselt. Equilibria in competitive location models. In H. A. Eiselt and V. Marianov, editors, *Foundations of Location Analysis*, chapter 7, pages 139–162. Springer-Verlag, Berlin, 2011.

H. A. Eiselt and V. Marianov, editors. *Foundations of Location Analysis*. Springer-Verlag, New York, 2011.

H. A. Eiselt and C.-L. Sandblom. *Decision Analysis, Location Models, and Scheduling Problems*. Springer, New York, 2004.

H. A. Eiselt, G. Laporte, and J.-F. Thisse. Competitive location models: A framework and bibliography. *Transportation Science*, 27(1):44–54, 1993.

S. Elloumi, M. Labbé, and Y. Pochet. A new formulation and resolution method for the p-center problem. *INFORMS Journal on Computing*, 16(1):84–94, 2004.

G. D. Eppen. Effects of centralization on expected costs in a multi-location newsboy problem. *Management Science*, 25(5):498–501, 1979.

R. Epstein, L. Henríquez, J. Catalán, G. Y. Weintraub, C. Martínez, and F. Espejo. A combinatorial auction improves school meals in Chile: A case of OR in developing countries. *International Transactions in Operational Research*, 11:593–612, 2004.

S. J. Erlebacher and R. D. Meller. The interaction of location and inventory in designing distribution systems. *IIE Transactions*, 32:155–166, 2000.

D. Erlenkotter. A dual-based procedure for uncapacitated facility location. *Operations Research*, 26 (6):992–1009, 1978.

D. Erlenkotter. Ford Whitman Harris and the economic order quantity model. *Operations Research*, 38(6):937–946, 1990.

I. Farasyn, K. Perkoz, and W. Van de Velde. Spreadsheet models for inventory target setting at Procter & Gamble. *Interfaces*, 38(4):241–250, 2008.

I. Farasyn, S. Humair, J. I. Kahn, J. J. Neale, O. Rosen, J. Ruark, W. Tarlton, W. Van de Velde, G. Wegryn, and S. P. Willems. Inventory optimization at Procter & Gamble: Achieving real benefits through user adoption of inventory tools. *Interfaces*, 41(1):66–78, 2011.

M. Farvid and K. Rosling. The square root algorithm for single item lot sizing. Working paper, Linnæus University, Växjö, Sweden, 2014.

M. H. Fazel Zarandi, S. Davari, and S. A. Haddad Sisakht. The large scale maximal covering location problem. *Scientia Iranica*, 18(6):1564–1570, 2011.

A. Federgruen and M. Tzur. A simple forward algorithm to solve general dynamic lot sizing models with n periods in $O(n \log n)$ or $O(n)$ time. *Management Science*, 37(8):909–925, 1991.

A. Federgruen and Y.-S. Zheng. An efficient algorithm for computing an optimal (r, Q) policy in continuous review stochastic inventory systems. *Operations Research*, 40(4):808–813, 1992.

A. Federgruen and P. Zipkin. Computational issues in an infinite-horizon, multiechelon inventory model. *Operations Research*, 32(2):818–836, 1984.

E. Feitzinger and H. L. Lee. Mass customization at Hewlett-Packard: The power of postponement. *Harvard Business Review*, 75:116–123, 1997.

K. J. Ferreira, B. H. A. Lee, and D. Simchi-Levi. Analytics for an online retailer: Demand forecasting and price optimization. *Manufacturing & Service Operations Management*, 18(1):69–88, 2015.

S. R. Finch. *Mathematical Constants*. Cambridge University Press, New York, 2003.

M. L. Fisher. The Lagrangian relaxation method for solving integer programming problems. *Management Science*, 27(1):1–18, 1981.

M. L. Fisher. An applications oriented guide to Lagrangian relaxation. *Interfaces*, 15(2):10–21, 1985.

M. L. Fisher. Optimal solution of vehicle routing problems using minimum k-trees. *Operations Research*, 42(4):626–642, 1994a.

M. L. Fisher. A polynomial algorithm for the degree-constrained minimum K-tree problem. *Operations Research*, 42(4):775–779, 1994b.

M. L. Fisher and R. Jaikumar. A generalized assignment heuristic for vehicle routing. *Networks*, 11 (2):109–124, 1981.

B. Fleischmann. A new class of cutting planes for the symmetric travelling salesman problem. *Mathematical Programming Series B*, 40(1–3):225–246, 1988.

M. M. Flood. The traveling-salesman problem. *Operations Research*, 4(1):61–75, 1956.

M. Florian, J. K. Lenstra, and A. H. G. Rinnooy Kan. Deterministic production planning: Algorithms and complexity. *Management Science*, 26(7):669–679, 1980.

L. R. Ford and D. R. Fulkerson. A suggested computation for maximal multi-commodity network flows. *Management Science*, 5(1):97–101, 1958.

J. W. Forrester. Industrial dynamics: A major breakthrough for decision makers. *Harvard Business Review*, 36(4):37–66, 1958.

B. A. Foster and D. M. Ryan. An integer programming approach to the vehicle scheduling problem. *Operational Research Quarterly*, 27(2, part 1):367–384, 1976.

P. Francis and K. Smilowitz. Modeling techniques for periodic vehicle routing problems. *Transportation Research Part B: Methodological*, 40(10):872–884, 2006.

P. Francis, K. Smilowitz, and M. Tzur. The period vehicle routing problem with service choice. *Transportation Science*, 40(4):439–454, 2006.

P. M. Francis, K. R. Smilowitz, and M. Tzur. The period vehicle routing problem and its extensions. In B. Golden, S. Raghavan, and E. Wasil, editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, volume 43 of *Operations Research/Computer Science Interfaces*, pages 73–102. Springer US, New York, 2008.

S. Frank, I. Steponavice, and S. Rebennack. Optimal power flow: A bibliographic survey I: Formulations and deterministic methods. *Energy Systems*, 3(3):221–258, 2012a.

S. Frank, I. Steponavice, and S. Rebennack. Optimal power flow: A bibliographic survey II: Nondeterministic and hybrid methods. *Energy Systems*, 3(3):259–289, 2012b.

J. Friedman, T. Hastie, and R. Tibshirani. *The Elements of Statistical Learning*. Springer, Berlin, 2001.

G. Gallego, O. Özer, and P. H. Zipkin. Bounds, heuristics, and approximations for distribution systems. *Operations Research*, 55(3):503–517, 2007.

H. P. Galliher, P. M. Morse, and M. Simond. Dynamics of two classes of continuous-review inventory systems. *Operations Research*, 7(3):362–384, 1959.

R. D. Galvão. A dual-bounded algorithm for the p-median problem. *Operations Research*, 28(5): 1112–1121, 1980.

R. D. Galvão and C. ReVelle. A Lagrangean heuristic for the maximal covering location problem. *European Journal of Operational Research*, 88:114–123, 1996.

M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, 1979.

R. S. Garfinkel, A. W. Neebe, and M. R. Rao. An algorithm for the m-median plant location problem. *Transportation Science*, 8(3):217–236, 1974.

L. Garrow. *Discrete Choice Modelling and Air Travel Demand: Theory and Applications*. Routledge, New York, 2010.

Gartner, Inc. Gartner announces rankings of the 2016 supply chain top 25, May 19, 2016. URL `http://www.gartner.com/newsroom/id/3323617`.

T. J. Gaskell. Bases for fleet scheduling. *Operational Research Quarterly*, 18:281–295, 1967.

B. Gavish and K. Altinkemer. Backbone network design tools with economic tradeoffs. *ORSA Journal on Computing*, 2(3):236–252, 1990.

S. Geary, S. M. Disney, and D. R. Towill. On bullwhip in supply chains—historical review, present practice and expected future impact. *International Journal of Production Economics*, 101(1): 2–18, 2006.

M. Gendreau and J.-Y. Potvin, editors. *Handbook of Metaheuristics*. Springer, New York, 2nd edition, 2010.

M. Gendreau, A. Hertz, and G. Laporte. New insertion and postoptimization procedures for the traveling salesman problem. *Operations Research*, 40(6):1086–1094, 1992.

M. Gendreau, A. Hertz, and G. Laporte. A tabu search heuristic for the vehicle routing problem. *Management Science*, 40(10):1276–1290, 1994.

M. Gendreau, G. Laporte, and J.-Y. Potvin. Metaheuristics for the capacitated VRP. In P. Toth and D. Vigo, editors, *The Vehicle Routing Problem*, chapter 6, pages 129–154. Society for Industrial and Applied Mathematics, Philadelphia, 2001.

M. Gendreau, J.-Y. Potvin, O. Bräumlaysy, G. Hasle, and A. Løkketangen. Metaheuristics for the vehicle routing problem and its extensions: A categorized bibliography. In B. Golden, S. Raghavan, and E. Wasil, editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, volume 43 of *Operations Research/Computer Science Interfaces*, pages 143–169. Springer US, New York, 2008.

B. Gendron, T. G. Crainic, and A. Frangioni. Multicommodity capacitated network design. In B. Sansò and P. Soriano, editors, *Telecommunications Network Planning*, chapter 1, pages 1–19. Springer, New York, 1999.

A. Geoffrion and R. McBride. Lagrangean relaxation applied to capacitated facility location problems. *AIIE Transactions*, 10(1):40–47, 1978.

A. M. Geoffrion and G. W. Graves. Multicommodity distribution system design by Benders decomposition. *Management Science*, 20(5):822–844, 1974.

I. Ghamlouche, T. G. Crainic, and M. Gendreau. Cycle-based neighbourhoods for fixed-charge capacitated multicommodity network design. *Operations Research*, 51(4):655–667, 2003.

A. Ghosh and C. S. Craig. Formulating retail location strategy in a changing environment. *The Journal of Marketing*, 47(3):56–68, 1983.

B. E. Gillett and L. R. Miller. A heuristic algorithm for the vehicle-dispatch problem. *Operations Research*, 22(2):340–349, 1974.

P. Gilmore and R. E. Gomory. Multistage cutting stock problems of two and more dimensions. *Operations Research*, 13(1):94–120, 1965.

F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13:533–549, 1986.

F. Glover. Tabu search—Part I. *ORSA Journal on Computing*, 1(3):190–206, 1989.

F. Glover. Tabu search—Part II. *ORSA Journal on Computing*, 2(1):4–32, 1990.

F. Glover and M. Laguna. *Tabu Search*. Kluwer, Boston, 1997.

M. X. Goemans. Worst-case comparison of valid inequalities for the TSP. *Mathematical Programming*, 69:335–349, 1995.

M. X. Goemans and D. P. Williamson. The primal-dual method for approximation algorithms and its application to network design problems. In D. S. Hochbaum, editor, *Approximation Algorithms for NP-Hard Problems*, chapter 4, pages 144–191. PWS Publishing Co., Boston, MA, 1997.

M. Goetschalckx and C. Jacobs-Blecha. The vehicle routing problem with backhauls. *European Journal of Operational Research*, 42:39–51, 1989.

D. A. Goldberg, D. A. Katz-Rogozhnikov, Y. Lu, M. Sharma, and M. S. Squillante. Asymptotic optimality of constant-order policies for lost sales inventory models with large lead times. *Mathematics of Operations Research*, 41(3):898–913, 2016.

D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA, 1989.

B. Golden, S. Raghavan, and E. Wasil, editors. *The Vehicle Routing Problem: Latest Advances and New Challenges*, volume 43 of *Operations Research/Computer Science Interfaces*. Springer US, New York, 2008.

B. L. Golden and W. R. Stewart. Empirical analysis of heuristics. In E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, editors, *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, chapter 7, pages 207–249. John Wiley & Sons, New York, 1985.

E. Gourdin, M. Labbé, and H. Yaman. Telecommunication and location. In Z. Drezner and H. W. Hamacher, editors, *Facility Location: Applications and Theory*, chapter 9, pages 277–307. Springer-Verlag, New York, 2002.

J. E. Graver and M. E. Watkins. *Combinatorics with Emphasis on the Theory of Graphs*. Springer-Verlag, New York, 1977.

S. C. Graves. A multi-echelon inventory model for a repairable item with one-for-one replenishment. *Management Science*, 31(10):1247–1256, 1985.

S. C. Graves. Safety stocks in manufacturing systems. *Journal of Manufacturing and Operations Management*, 1:67–101, 1988.

S. C. Graves and T. Schoenmeyr. Strategic safety-stock placement in supply chains with capacity constraints. *Manufacturing & Service Operations Management*, 18(3):445–460, 2016.

S. C. Graves and B. T. Tomlin. Process flexibility in supply chains. *Management Science*, 49(7): 907–919, 2003.

S. C. Graves and S. P. Willems. Optimizing strategic safety stock placement in supply chains. *Manufacturing & Service Operations Management*, 2(1):68–83, 2000.

S. C. Graves and S. P. Willems. Supply chain design: Safety stock placement and supply chain configuration. In A. G. de Kok and S. C. Graves, editors, *Supply Chain Management: Design, Coordination and Operation*, volume 11 of *Handbooks in Operations Research and Management Science*, chapter 3. Elsevier, Amsterdam, 2003a.

S. C. Graves and S. P. Willems. Erratum: Optimization strategic safety stock placement in supply chains. *Manufacturing & Service Operations Management*, 5(2):176–177, 2003b.

A. Grosfeld Nir and Y. Gerchak. Multiple lotsizing in production to order with random yields: Review of recent advances. *Annals of Operations Research*, 126(1–4):43–69, 2004.

M. Grötschel. On the monotone symmetric travelling salesman problem: Hypohamiltonian/hypotraceable graphs and facets. *Mathematics of Operations Research*, 5(2):285–292, 1980a.

M. Grötschel. On the symmetric travelling salesman problem: Solution of a 120-city problem. In M. W. Padberg, editor, *Combinatorial Optimization*, volume 12 of *Mathematical Programming Studies*, pages 61–77. Springer, Berlin, 1980b.

M. Grötschel and O. Holland. A cutting plane algorithm for minimum perfect 2-matchings. *Computing*, 39(4):327–344, 1987.

M. Grötschel and O. Holland. Solution of large-scale symmetric travelling salesman problems. *Mathematical Programming Series B*, 51(1–3):141–202, 1991.

M. Grötschel and M. W. Padberg. On the symmetric travelling salesman problem II: Lifting theorems and facets. *Mathematical Programming Series B*, 16(1):281–302, 1979.

M. Grötschel and W. R. Pulleyblank. Clique tree inequalities and the symmetric travelling salesman problem. *Mathematics of Operations Research*, 11(4):537–569, 1986.

M. Guignard and S. Kim. Lagrangean decomposition: A model yielding strong Lagrangean bounds. *Mathematical Programming*, 39:215–228, 1987.

R. Güllü, E. Onol, and N. Erkip. Analysis of a deterministic demand production/inventory system under nonstationary supply uncertainty. *IIE Transactions*, 29(8):703–709, 1997.

R. Güllü, E. Önol, and N. Erkip. Analysis of an inventory system under supply uncertainty. *International Journal of Production Economics*, 59:377–385, 1999.

G. Gunawardane. Dynamic versions of set covering type public facility location problems. *European Journal of Operational Research*, 10(2):190–195, 1982.

O. Günlük. A branch-and-cut algorithm for capacitated network design problems. *Mathematical Programming*, 86(1):17–39, 1999.

E. A. Gunn, C. A. MacDonald, A. Friars, and G. Caissie. Scotsburn Dairy Group uses a hierarchical production scheduling and inventory management system to control its ice cream production. *Interfaces*, 44(3):253–268, 2014.

G. Hadley and T. M. Whitin. *Analysis of Inventory Systems*. Prentice-Hall, Englewood Cliffs, NJ, 1963.

M. Haimovich and A. H. G. Rinnooy Kan. Bounds and heuristics for capacitated routing problems. *Mathematics of Operations Research*, 10(4):527–542, 1985.

S. L. Hakimi. Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Operations Research*, 13(3):462–475, 1965.

S. L. Hakimi. On locating new facilities in a competitive environment. *European Journal of Operational Research*, 12:29–55, 1983.

J. Han, Y. Xu, D. Liu, Y. Zhao, Z. Zhao, S. Zhou, T. Deng, M. Xue, J. Ye, and Z.-J. M. Shen. Operations research enables better planning of natural gas pipelines. *INFORMS Journal on Applied Analytics*, 49(1):23–39, 2019.

P. Hansen and N. Mladenović. Variable neighborhood search for the p-median. *Location Science*, 5 (4):207–226, 1997.

R. Hariharan and P. Zipkin. Customer-order information, leadtimes, and inventories. *Management Science*, 41(10):1599–1607, 1995.

F. W. Harris. How many parts to make at once. *Factory: The Magazine of Management*, 10(2):135, 1913. Reprinted in *Operations Research* 38(6):947–950, 1990.

P. Harsha and M. Dahleh. Optimal management and sizing of energy storage under dynamic pricing for the efficient integration of renewable energy. *IEEE Transactions on Power Systems*, 30(3): 1164–1181, 2015.

R. M. Harstad and A. S. Pekeč. Relevance to practice and auction theory: A memorial essay for Michael Rothkopf. *Interfaces*, 38(5):367–380, 2008.

M. Hartman, A. B. Martin, N. Espinosa, A. Catlin, and The National Health Expenditure Accounts Team. National health care spending in 2016: Spending and enrollment growth slow after initial coverage expansions. *Health Affairs*, 37(1):150–160, 2018.

A. C. Hax and D. Candea. *Production and Inventory Management*. Prentice-Hall, Englewood Cliffs, NJ, 1984.

M. Held and R. M. Karp. A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied Mathematics*, 10(1):196–210, 1962.

M. Held and R. M. Karp. The traveling-salesman problem and minimum spanning trees. *Operations Research*, 18(6):1138–1162, 1970.

M. Held and R. M. Karp. The traveling-salesman problem and minimum spanning trees: Part II. *Mathematical Programming*, 1(1):6–25, 1971.

J. Henry. BMW says flexible, not lean, is the next big thing in autos. *BNET*, November 24, 2009.

Y. T. Herer, M. Tzur, and E. Yücesan. The multi-location transshipment problem. *IIE Transactions*, 38(3):185–200, 2006.

J. L. Heskett and S. Signorelli. Benetton (A). Harvard Business School Case, September 13, 1984.

I. Higa, A. M. Feyerherm, and A. L. Machado. Waiting time in an $(S - 1, S)$ inventory system. *Operations Research*, 23(4):674–680, 1975.

T. H. Ho, S. Savin, and C. Terwiesch. Managing demand and sales dynamics in new product diffusion under supply constraint. *Management Science*, 48(2):187–206, 2002.

K. L. Hoffman, M. Padberg, and G. Rinaldi. Traveling salesman problem. In S. I. Gass and M. C. Fu, editors, *Encyclopedia of Operations Research and Management Science*, pages 1573–1578. Springer US, New York, 2013.

G. Hohner, J. Rich, E. Ng, G. Reid, A. J. Davenport, J. R. Kalagnanam, H. S. Lee, and C. An. Combinatorial and quantity-discount procurement auctions benefit Mars, Incorporated and its suppliers. *Interfaces*, 33(1):23–35, 2003.

C. Holland, J. Levis, R. Nuggehalli, B. Santilli, and J. Winters. UPS optimizes delivery routes. *Interfaces*, 47(1):8–23, 2017.

J. H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. University of Michigan Press, Ann Arbor, MI, 1992.

O. Holland. *Schnittebenenverfahren für Travelling-Salesman und verwandte Probleme*. Ph.D. dissertation, Universität Bonn, Bonn, Germany, 1987.

K. Holmberg and D. Yuan. A Lagrangian heuristic based branch-and-bound approach for the capacitated network design problem. *Operations Research*, 48(3):461–481, 2000.

C. C. Holt. Forecasting seasonal and trends by exponentially weighted moving averages. Office of Naval Research Memorandum, No. 52, 1957.

S. Hong. *A Linear Programming Approach for the Traveling Salesman Problem*. Ph.D. dissertation, Johns Hopkins University, Baltimore, MD, 1972.

C. M. Hosage and M. F. Goodchild. Discrete space location-allocation solutions from genetic algorithms. *Annals of Operations Research*, 6(2):35–46, 1986.

H. Hotelling. Stability in competition. *The Economic Journal*, 39(153):41–57, 1929.

M. Huang, K. Smilowitz, and B. Balcik. Models for relief routing: Equity, efficiency and efficacy. *Transportation Research Part E: Logistics and Transportation Review*, 48(1):2–18, 2012.

W. T. Huh, G. Janakiraman, J. A. Muckstadt, and P. Rusmevichientong. Asymptotic optimality of order-up-to policies in lost sales inventory systems. *Management Science*, 55(3):404–420, 2009.

S. Humair and S. P. Willems. Technical note: Optimizing strategic safety stock placement in general acyclic networks. *Operations Research*, 59(3):781–787, 2011.

S. Humair, J. D. Ruark, B. Tomlin, and S. P. Willems. Incorporating stochastic lead times into the guaranteed service model of safety stock optimization. *Interfaces*, 43(5):421–434, 2013.

T. Ida and T. Kuroda. Discrete choice model analysis of demand for mobile telephone service in Japan. *Empirical Economics*, 36:65–80, 2009.

K. Inderfurth. Safety stock optimization in multi-stage inventory systems. *International Journal of Production Economics*, 24:103–113, 1991.

K. Inderfurth and S. Minner. Safety stocks in multi-stage inventory systems under different service measures. *European Journal of Operational Research*, 106:57–73, 1998.

INFORMS. MISO applies operations research to energy and ancillary services markets unlocking billions of saving, 2011. URL `https://www.informs.org/Impact/O.R.-Analytics-Success-Stories/MISO-Applies-O.R.-to-Energy-and-Ancillary-Services-Markets`.

INFORMS. 2018 INFORMS Franz Edelman Award finalists selected from leading analytics teams around the world, 2018. URL `https://www.informs.org/About-INFORMS/News-Room/Press-Releases/2018-INFORMS-Franz-Edelman-Award-finalists-selected-from-leading-analytics-teams-around-the-world`.

T. Islam. Household level innovation diffusion model of photo-voltaic (PV) solar cells from stated preference data. *Energy Policy*, 65(Supplement C):340–350, 2014.

G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning with Applications in R*. Springer, New York, 2013.

J. H. Jaramillo, J. Bhadury, and R. Batta. On the use of genetic algorithms to solve location problems. *Computers & Operations Research*, 29(6):761–779, 2002.

M. Jeger and O. Kariv. Algorithms for finding P-centers on a weighted tree (for relatively small P). *Networks*, 15(3):381–389, 1985.

D. S. Johnson. Local optimization and the traveling salesman problem. In *Proceedings of the 17th Colloquium on Automata, Languages, and Programming*, pages 446–461, Berlin, 1995.

D. S. Johnson and L. A. McGeoch. The traveling salesman problem: A case study in local optimization. In E. H. L. Aarts and J. K. Lenstra, editors, *Local Search in Combinatorial Optimization*, pages 215–310. John Wiley & Sons, New York, 1997.

M. E. Johnson, H. L. Lee, T. Davis, and R. Hall. Expressions for item fill rates in periodic inventory systems. *Naval Research Logistics*, 42:57–80, 1995.

M. P. Johnson. Single-period location models for subsidized housing: Project-based subsidies. *Socio-Economic Planning Sciences*, 40:249–274, 2006.

M. P. Johnson and K. Smilowitz. Community-based operations research. In T. Klastorin, editor, *Tutorials in Operations Research*, chapter 6, pages 102–123. INFORMS, Hanover, MD, 2007.

W. C. Jordan and S. C. Graves. Principles on the benefits of manufacturing process flexibility. *Management Science*, 41(4):577–594, 1995.

M. Jünger and W. R. Pulleyblank. Geometric duality and combinatorial optimization. In S. D. Chatterji, B. Fuchssteiner, and U. Kulisch, editors, *Jahrbuch Überblicke Mathematik*, pages 1–24. Vieweg, Brundschweig/Wiesbaden, Germany, 1993.

M. Jünger, M. Schulz, and W. Zychowicz. GEODUAL. Software, 2009. URL `https://informatik.uni-koeln.de/ls-juenger/geodual`.

Kaggle.com. Retail data analytics [data set], September 2017. URL `https://www.kaggle.com/manjeetsingh/retaildataset`.

J. Kalagnanam and D. C. Parkes. Auctions, bidding and exchange design. In D. Simchi-Levi, S. D. Wu, and Z.-J. M. Shen, editors, *Handbook of Quantitative Supply Chain Analysis: Modeling in the E-Business Era*, chapter 5. Springer, New York, 2004.

L. V. Kantorovich. Mathematical methods of organizing and planning production. *Management Science*, 6(4):366–422, 1960.

R. Kapuscinski, R. Q. Zhang, P. Carbonneau, R. Moore, and B. Reeves. Inventory decisions in Dell's supply chain. *Interfaces*, 34(3):191–205, 2004.

B. Y. Kara and M. R. Taner. Hub location problems: The location of interacting facilities. In H. A. Eiselt and V. Marianov, editors, *Foundations of Location Analysis*, chapter 12, pages 273–288. Springer-Verlag, Berlin, 2011.

R. L. Karg and G. L. Thompson. A heuristic approach to solving travelling salesman problems. *Management Science*, 10(2):225–248, 1964.

O. Kariv and S. L. Hakimi. An algorithmic approach to network location problems. I: The p-centers. *SIAM Journal on Applied Mathematics*, 37(3):513–538, 1979a.

O. Kariv and S. L. Hakimi. An algorithmic approach to network location problems. II: The p-medians. *SIAM Journal on Applied Mathematics*, 37(3):539–560, 1979b.

S. Karlin. Optimal inventory policy for the Arrow-Harris-Marschak dynamic model. In K. J. Arrow, S. Karlin, and H. Scarf, editors, *Studies in the Mathematical Theory of Inventory and Production*, chapter 9, pages 135–154. Stanford University Press, Stanford, CA, 1958a.

S. Karlin. One stage inventory models with uncertainty. In K. J. Arrow, S. Karlin, and H. Scarf, editors, *Studies in the Mathematical Theory of Inventory and Production*, chapter 8, pages 109–134. Stanford University Press, Stanford, CA, 1958b.

S. Karlin. Dynamic inventory policy with varying stochastic demands. *Management Science*, 6(3): 231–258, 1960.

S. Karlin and H. Scarf. Inventory models of the Arrow-Harris-Marschak type with time lag. In K. J. Arrow, S. Karlin, and H. Scarf, editors, *Studies in the Mathematical Theory of Inventory and Production*, chapter 10, pages 155–178. Stanford University Press, Stanford, CA, 1958.

R. M. Karp and J. M. Steele. Probabilistic analysis of heuristics. In E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, editors, *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, chapter 6, pages 181–205. John Wiley & Sons, Hoboken, NJ, 1985.

M. Karpinski, M. Lampis, and R. Schmied. New inapproximability bounds for TSP. In L. Cai, S.-W. Cheng, and T.-W. Lam, editors, *Algorithms and Computation*, volume 8283 of *Lecture Notes in Computer Science*, pages 568–578. Springer, Berlin, 2013.

T. T. Ke, Z.-J. M. Shen, and S. Li. How inventory cost influences introduction timing of product line extensions. *Production and Operations Management*, 22(5):1214–1231, 2013.

J. Khawam, W. H. Hausman, and D. W. Cheng. Warranty inventory optimization for Hitachi Global Storage Technologies, Inc. *Interfaces*, 37(5):455–471, 2007.

J. H. Kim and W. B. Powell. Optimal energy commitments with storage and intermittent supply. *Operations Research*, 59(6):1347–1360, 2011.

G. E. Kimball. General principles of inventory control. *Journal of Manufacturing and Operations Management*, 1:119–130, 1988.

J. G. Klincewicz and H. Luss. A Lagrangian relaxation heuristic for capacitated facility location with single-source constraints. *Journal of the Operational Research Society*, 37(5):495–500, 1986.

N. Kohl. *Exact Methods for Time Constrained Routing and Related Scheduling Problems*. Ph.D. dissertation, Technical University of Denmark, Lyngby, Denmark, 1995.

N. Kohl and O. B. G. Madsen. An optimization algorithm for the vehicle routing problem with time windows based on Lagrangian relaxation. *Operations Research*, 45(3):395–406, 1997.

F. Koppelman. Travel prediction with models of individualistic choice behavior. Technical report, Department of Civil Engineering, MIT, Cambridge, MA, 1975.

M. Körkel. On the exact solution of large-scale simple plant location problems. *European Journal of Operational Research*, 39(2):157–173, 1989.

M. Korpaas, A. T. Holen, and R. Hildrum. Operation and sizing of energy storage for wind power plants in a market system. *Electrical Power and Energy Systems*, 25:599–606, 2003.

K. Krishnan and V. R. K. Rao. Inventory control in N warehouses. *Journal of Industrial Engineering*, 16:212–215, 1965.

J. B. Kruskal, Jr. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, 1956.

M. J. Kuby. Programming models for facility dispersion: The p-dispersion and maxisum dispersion problems. *Geographical Analysis*, 19:315–329, 1987.

A. A. Kuehn and M. J. Hamburger. A heuristic program for locating warehouses. *Management Science*, 9(4):643–666, 1963.

S. Kunnumkal and H. Topaloglu. A duality-based relaxation and decomposition approach for inventory distribution systems. *Naval Research Logistics*, 55(7):612–631, 2008.

Y.-J. Kuo and H. D. Mittelmann. Interior point methods for second-order cone programming and OR applications. *Computational Optimization and Applications*, 28(3):255–285, 2004.

A. A. Kurawarwala and H. Matsuo. Forecasting and inventory management of short life-cycle products. *Operations Research*, 44(1):131–150, 1996.

G. Laporte. The traveling salesman problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(2):231–247, 1992a.

G. Laporte. The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(3):345–358, 1992b.

G. Laporte. A concise guide to the traveling salesman problem. *Journal of the Operational Research Society*, 61:35–40, 2010.

G. Laporte and Y. Nobert. An exact algorithm for minimizing routing and operating cost in depot location. *European Journal of Operational Research*, 6(2):224–226, 1981.

G. Laporte and Y. Nobert. Comb inequalities for the vehicle routing problem. *Methods of Operations Research*, 51:271–276, 1984.

G. Laporte and Y. Nobert. Exact algorithms for the vehicle routing problem. In S. Martello, G. Laporte, M. Minoux, and C. Ribeiro, editors, *Surveys in Combinatorial Optimization*, volume 132 of *North-Holland Mathematics Studies*, pages 147–184. North-Holland, Amsterdam, 1987.

G. Laporte and F. Semet. Classical heuristics for the capacitated VRP. In P. Toth and D. Vigo, editors, *The Vehicle Routing Problem*, chapter 5, pages 109–128. Society for Industrial and Applied Mathematics, Philadelphia, 2001.

G. Laporte, Y. Nobert, and M. Desrochers. Optimal routing under capacity and distance restrictions. *Operations Research*, 33(5):1050–1073, 1985.

G. Laporte, Y. Nobert, and D. A. Arpin. An exact algorithm for solving a capacitated location-routing problem. *Annals of Operations Research*, 6(9):291–310, 1986.

G. Laporte, M. Gendreau, J.-Y. Potvin, and F. Semet. Classical and modern heuristics for the vehicle routing problem. *International Transactions in Operational Research*, 7(4–5):285–300, 2000.

M. A. Lariviere. A note on probability distributions with increasing generalized failure rates. *Operations Research*, 54(3):602–604, 2006.

M. A. Lariviere and E. L. Porteus. Selling to the newsvendor: An analysis of price-only contracts. *Manufacturing & Service Operations Management*, 3(4):293–305, 2001.

E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, editors. *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. John Wiley & Sons, Hoboken, NJ, 1985.

D. Lawrence. Bridging the quality chasm. In P. P. Reid, W. D. Compton, J. H. Grossman, and G. Fanjiang, editors, *Building a Better Delivery System: A New Engineering/Health Care Partnership*. National Academies Press, Washington, DC, 2005.

F.-X. Le Louarn, M. Gendreau, and J.-Y. Potvin. GENI ants for the traveling salesman problem. *Annals of Operations Research*, 131(1–4):187–201, 2004.

H. L. Lee. Effective inventory and service management through product and process redesign. *Operations Research*, 44(1):151–159, 1996.

H. L. Lee and C. Billington. Material management in decentralized supply chains. *Operations Research*, 41(5):835–847, 1993.

H. L. Lee, V. Padmanabhan, and S. Whang. Information distortion in a supply chain: The bullwhip effect. *Management Science*, 43(4):546–558, 1997a.

H. L. Lee, V. Padmanabhan, and S. Whang. The bullwhip effect in supply chains. *Sloan Management Review*, 38(3):93–102, 1997b.

H. L. Lee, V. Padmanabhan, and S. Whang. Comments on "Information distortion in a supply chain: The bullwhip effect". *Management Science*, 50(12S):1887–1893, 2004.

E. Lesnaia. *Optimizing Safety Stock Placement in General Network Supply Chains*. Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, September 2004.

A. N. Letchford, G. Reinelt, and D. O. Theis. A faster exact separation algorithm for blossom inequalities. In D. Bienstock and G. Nemhauser, editors, *Integer Programming and Combinatorial Optimization*, volume 3064 of *Lecture Notes in Computer Science*. Springer, Berlin, 2004.

R. Levi, G. Janakiraman, and M. Nagarajan. A 2-approximation algorithm for stochastic inventory control models with lost sales. *Mathematics of Operations Research*, 33(2):351–374, 2008.

C.-L. Li and D. Simchi-Levi. Worst-case analysis of heuristics for multidepot capacitated vehicle routing problems. *ORSA Journal on Computing*, 2(1):64–73, 1990.

X. Li and Y. Ouyang. A continuum approximation approach to reliable facility location design under correlated probabilistic disruptions. *Transportation Research–Part B*, 44(4):535–548, 2010.

R. W. Lien, S. M. R. Iravani, K. Smilowitz, and M. Tzur. An efficient and robust design for transshipment networks. *Production and Operations Management*, 20(5):699–713, 2011.

G. L. Lilien and A. Rangaswamy. *Marketing Engineering: Computer-Assistend Marketing Analysis and Planning*. Addison-Wesley Educational Publishers, Reading, MA, 1998.

G. L. Lilien, A. Rangaswamy, and A. D. Bruyn. *Principles of Marketing Engineering*. Trafford Publishing, Victoria, BC, Canada, 2007.

M. Lim, A. Bassamboo, S. Chopra, and M. S. Daskin. Use of chaining strategies in the presence of disruption risks. Working paper, University of Illinois, Champaign, 2012.

S. Lin. Computer solutions of the traveling salesman problem. *Bell System Technical Journal*, 44: 2245–2269, 1965.

S. Lin and B. W. Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, 21:498–516, 1973.

K. Linebaugh. Honda's flexible plants provide edge. *Wall Street Journal*, September 23, 2008.

J. D. C. Little, K. G. Murty, D. W. Sweeney, and C. Karel. An algorithm for the traveling salesman problem. *Operations Research*, 11(6):972–989, 1963.

M. E. Lübbecke and J. Desrosiers. Selected topics in column generation. *Operations Research*, 53 (6):1007–1023, 2005.

S. Luke. *Essentials of Metaheuristics*. Lulu, 2nd edition, 2013. URL `http://cs.gmu.edu/~sean/book/metaheuristics`.

T. L. Magnanti and R. T. Wong. Network design and transportation planning: Models and algorithms. *Transportation Science*, 18(1):1–55, 1984.

T. L. Magnanti, P. Mireault, and R. T. Wong. Tailoring Benders decomposition for uncapacitated network design. *Mathematical Programming Study*, 26:112–154, 1986.

T. L. Magnanti, Z.-J. M. Shen, J. Shu, D. Simchi-Levi, and C.-P. Teo. Inventory placement in acyclic supply chain networks. *Operations Research Letters*, 34:228–238, 2006.

V. Mahajan, E. Muller, and F. M. Bass. Diffusion of new products: Empirical generalizations and managerial uses. *Marketing Science*, 14(3S):G79–G88, 1995.

H. Mak and Z.-J. M. Shen. Stochastic programming approach to process flexibility design. *Flexible Services and Manufacturing Journal*, 21(3):75–91, 2009.

A. S. Manne. Plant location under economies-of-scale—decentralization and computation. *Management Science*, 11(2):213–235, 1964.

F. E. Maranzana. On the location of supply points to minimize transport costs. *Operational Research Quarterly*, 15(3):261–270, 1964.

O. Martin, S. W. Otto, and E. W. Felten. Large-step Markov chains for the traveling salesman problem. *Complex Systems*, 5:299–326, 1991.

A. McAfee and E. Brynjolfsson. Big data: The management revolution. *Harvard Business Review*, 90(12):16–17, 2012.

S. T. McCormick, M. R. Rao, and G. Rinaldi. Easy and difficult objective functions for max cut. *Mathematical Programming*, 94(2–3):459–466, 2003.

P. McCullen and D. Towill. Diagnosis and reduction of bullwhip in supply chains. *Supply Chain Management*, 7(3):164–179, 2002.

D. McCutcheon. Flexible manufacturing: IBM's Bromont semiconductor packaging plant. *Canadian Electronics*, 19(7):26, 2004.

D. L. McFadden. *Conditional Logit Analysis of Qualitative Choice Behavior*. Academic Press, New York, 1974.

D. L. McFadden. Econometric analysis of qualitative response models. In *Handbook of Econometrics*, volume 2. Elsevier Science Publishers BV, New York, 1984.

L. A. McLay. Discrete optimization models for homeland security and disaster management. In D. Aleman and A. Thiele, editors, *INFORMS Tutorials in Operations Research*, pages 111–132. INFORMS, Hanover, MD, 2015.

L. A. McLay, S. H. Jacobson, and J. E. Kobza. A multilevel passenger screening problem for aviation security. *Naval Research Logistics*, 53:183–197, 2006.

L. A. McLay, S. H. Jacobson, and A. G. Nikolaev. A sequential stochastic passenger screening problem for aviation security. *IIE Transactions*, 41(6):575–591, 2009.

L. A. McLay, A. J. Lee, and S. H. Jacobson. Risk-based policies for airport security checkpoint screening. *Transportation Science*, 44(3):333–349, 2010.

N. Megiddo, A. Tamir, E. Zemel, and R. Chandrasekaran. An $O(n \log^2 n)$ algorithm for the kth longest path in a tree with applications to location problems. *SIAM Journal on Computing*, 10 (2):328–337, 1981.

N. Megiddo, E. Zemel, and S. L. Hakimi. The maximum coverage location problem. *SIAM Journal of Algebraic and Discrete Methods*, 4(2):253–261, 1983.

M. J. Meixell and S. D. Wu. Scenario analysis of demand in a technology market using leading indicators. *IEEE Transactions on Semiconductor Manufacturing*, 14(1):65–75, 2001.

E. Melachrinoudis. The location of undesirable facilities. In H. A. Eiselt and V. Marianov, editors, *Foundations of Location Analysis*, chapter 10, pages 207–239. Springer-Verlag, Berlin, 2011.

C. E. Miller, A. W. Tucker, and R. A. Zemlin. Integer programming formulation of traveling salesman problems. *Journal of the ACM*, 7(4):326–329, 1960.

D. L. Miller. A matching based exact algorithm for capacitated vehicle routing problems. *ORSA Journal on Computing*, 7(1):1–9, 1995.

D. L. Miller and J. F. Pekny. A staged primal-dual algorithm for perfect b-matching with edge capacities. *ORSA Journal on Computing*, 7(3):298–320, 1995.

A. Mingozzi, S. Giorgi, and R. Baldacci. An exact method for the vehicle routing problem with backhauls. *Transportation Science*, 33:315–329, 1999.

E. Minieka. The m-center problem. *SIAM Review*, 12(1):138–139, 1970.

S. Minner. Dynamic programming algorithms for multi-stage safety stock optimization. *Operations Research Spektrum*, 19(4):261–271, 1997.

P. B. Mirchandani. Locational decisions on stochastic networks. *Geographical Analysis*, 12(2): 172–183, 1980.

P. B. Mirchandani and R. L. Francis, editors. *Discrete Location Theory*. Wiley-Interscience, New York, 1990.

P. B. Mirchandani, A. Oudjit, and R. T. Wong. “Multidimensional” extensions and a nested dual approach for the m-median problem. *European Journal of Operational Research*, 21(1): 121–137, 1985.

A. Mirzaian. Lagrangian relaxation for the star-star concentrator location problem: Approximation algorithm and bounds. *Networks*, 15:1–20, 1985.

N. Mladenović and P. Hansen. Variable neighborhood search. *Computers and Operations Research*, 24(11):1097–1100, 1997.

N. Mladenovic, J. Brimberg, P. Hansen, and J. A. Moreno-Pérez. The p-median problem: A survey of metaheuristic approaches. *European Journal of Operational Research*, 179(3):927–939, 2007.

J. G. Morris. On the extent to which certain fixed-charge depot location problems can be solved by LP. *Journal of the Operational Research Society*, 29(1):71–76, 1978.

P. Morse and G. Kimball. *Methods of Operations Research*. Technology Press of MIT, Cambridge, MA, 1951.

T. E. Morton. Bounds on the solution of the lagged optimal inventory equation with no demand backlogging and proportional costs. *SIAM Review*, 11(4):572–596, 1969.

M. Mourgaya and F. Vanderbeck. Column generation based heuristic for tactical planning in multi-period vehicle routing. *European Journal of Operational Research*, 183(3):1028–1041, 2007.

J. A. Muckstadt and R. O. Roundy. Analysis of multistage production systems. In S. C. Graves, A. H. G. Rinnooy Kan, and P. H. Zipkin, editors, *Handbooks in Operations Research and Management Science, vol. 4: Logistics of Production and Inventory*, chapter 2, pages 59–131. Elsevier, New York, 1993.

R. B. Myerson and M. A. Satterthwaite. Efficient mechanisms for bilateral trading. *Journal of Economic Theory*, 29(2):265–281, 1983.

D. Naddef and G. Rinaldi. The graphical relaxation: A new framework for the symmetric traveling salesman polytope. *Mathematical Programming*, 58(1):53–88, 1993.

D. Naddef and G. Rinaldi. Branch-and-cut algorithms for the capacitated VRP. In P. Toth and D. Vigo, editors, *The Vehicle Routing Problem*, chapter 3, pages 53–84. Society for Industrial and Applied Mathematics, Philadelphia, 2001.

Y. Nagata and O. Bräysy. Edge assembly-based memetic algorithm for the capacitated vehicle routing problem. *Networks*, 54(4):205–215, 2009.

Y. Nagata and S. Kobayashi. Edge assembly crossover: A high-power genetic algorithm for the traveling salesman problem. In *Proceedings of the 7th International Conference on Genetic Algorithms*, pages 450–457, East Lansing, MI, 1997. Morgan Kaufmann Publishers.

G. Nagy and S. Salhi. Location-routing: Issues, models and methods. *European Journal of Operational Research*, 177(2):649–672, 2007.

S. Nahmias. Optimal ordering policies for perishable inventory—II. *Operations Research*, 23(4): 735–749, 1975.

S. Nahmias. Simple approximations for a variety of dynamic leadtime lost-sales inventory models. *Operations Research*, 27(5):904–924, 1979.

S. Nahmias. *Production and Operations Analysis*. McGraw-Hill/Irwin, New York, 5th edition, 2005.

S. Nahmias and W. P. Pierskalla. Optimal ordering policies for a product that perishes in two periods subject to stochastic demand. *Naval Research Logistics Quarterly*, 20:207–229, 1973.

P. Nandakumar and T. E. Morton. Near myopic heuristics for the fixed-life perishability problem. *Management Science*, 39(12):1490–1498, 1993.

H. Naseraldin and Y. T. Herer. Integrating the number and location of retail outlets on a line with replenishment decisions. *Management Science*, 54(9):1666–1683, 2008.

National Academy of Engineering (US) and Institute of Medicine (US) Committee on Engineering and the Health Care System. A new partnership between systems engineering and health care. In P. P. Reid, W. D. Compton, J. H. Grossman, and G. Fanjiang, editors, *Building a Better Delivery System: A New Engineering/Health Care Partnership*, chapter 1. National Academies Press, Washington, DC, 2005.

National Fire Protection Association, Inc. NPFA 1710: Standard for the organization and deployment of fire suppression operations, emergency medical operations, and special operations to the public by career fire departments. Quincy, MA, 2001.

National Oceanic and Atmospheric Administration (NOAA). Climate data online [data set], 2017. URL `https://www.ncdc.noaa.gov/cdo-web`.

R. M. Nauss. An improved algorithm for the capacitated facility location problem. *Journal of the Operational Research Society*, 29(12):1195–1201, 1978.

R. M. Nauss and R. E. Markland. Theory and application of an optimizing procedure for lock box location analysis. *Management Science*, 27:855–865, 1981.

A. G. Nikolaev, S. H. Jacobson, and L. A. McLay. A sequential stochastic security system design problem for aviation security. *Transportation Science*, 41(2):182–194, 2007.

J. P. Norback and R. F. Love. Geometric approaches to solving the traveling salesman problem. *Management Science*, 23(11):1208–1223, 1977.

L. K. Nozick and M. A. Turnquist. Inventory, transportation, service quality and the location of distribution centers. *European Journal of Operational Research*, 129:362–371, 2001a.

L. K. Nozick and M. A. Turnquist. A two-echelon inventory allocation and distribution center location analysis. *Transportation Research Part E*, 37:421–441, 2001b.

NYC OpenData. Fire incident dispatch data [data set], October 2, 2017. URL `https://data.cityofnewyork.us/Public-Safety/Fire-Incident-Dispatch-Data/8m42-w767`.

M. E. O'Kelly. A quadratic integer program for the location of interacting hub facilities. *European Journal of Operational Research*, 32:393–404, 1987.

G. C. Oliveira, A. P. C. Costa, and S. Binato. Large scale transmission network planning using optimization and heuristic techniques. *IEEE Transactions on Power Systems*, 10(4):1828–1834, 1995.

H. L. Ong and J. Moore. Worst-case analysis of two travelling salesman heuristics. *Operations Research Letters*, 2(6):273–277, 1984.

I. Or. *Travelling Salesman-Type Combinatorial Problems and Their Relation to the Logistics of Blood-Banking*. Ph.D. dissertation, Northwestern University, Evanston, IL, 1976.

M. J. Osborne. *An Introduction to Game Theory*. Oxford University Press, New York, 2003.

I. H. Osman. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*, 41(4):421–451, 1993.

S. H. Owen and M. S. Daskin. Strategic facility location: A review. *European Journal of Operational Research*, 111(3):423–447, 1998.

O. Özer and H. Xiong. Stock positioning and performance estimation for distribution systems with service constraints. *IIE Transactions*, 40(12):1141–1157, 2008.

L. Ozsen, C. R. Coullard, and M. S. Daskin. Capacitated warehouse location model with risk pooling. *Naval Research Logistics*, 55(4):295–312, 2008.

M. Padberg and S. Hong. On the symmetric travelling salesman problem: A computational study. *Mathematical Programming Study*, 12:78–107, 1980.

M. Padberg and G. Rinaldi. Optimization of a 532-city symmetric traveling salesman problem by branch and cut. *Operations Research Letters*, 6(1):1–7, 1987.

M. Padberg and G. Rinaldi. An efficient algorithm for the minimum capacity cut problem. *Mathematical Programming Series B*, 47(1–3):19–36, 1990a.

M. Padberg and G. Rinaldi. Facet identification for the symmetric traveling salesman polytope. *Mathematical Programming Series B*, 47(1–3):219–257, 1990b.

M. Padberg and G. Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Review*, 33(1):60–100, 1991.

M. W. Padberg and M. R. Rao. Odd minimum cut-sets and b-matchings. *Mathematics of Operations Research*, 7(1):67–80, 1982.

N. Padhy. Unit commitment—a bibliographical survey. *IEEE Transactions on Power Systems*, 19 (2):1196–1205, 2004.

C. H. Papadimitriou. The Euclidean travelling salesman problem is NP-complete. *Theoretical Computer Science*, 4(3):237–244, 1977.

M. Parlar and D. Berkin. Future supply uncertainty in EOQ models. *Naval Research Logistics*, 38 (1):107–121, 1991.

B. A. Pasternack. Optimal pricing and return policies for perishable commodities. *Marketing Science*, 4(2):166–176, 1985.

A. Paul. A note on closure properties of failure rate distributions. *Operations Research*, 53(4): 733–734, 2005.

E. Paz-Frankel. Truck driver turnover reaches record level. *Memphis Business Journal*, April 2, 2006.

G. Perakis and G. Roels. The price of anarchy in supply chains: Quantifying the efficiency of price-only contracts. *Management Science*, 53(8):1249–1268, 2007.

J. Perl and M. S. Daskin. A warehouse location-routing problem. *Transportation Research Part B*, 19B(5):381–396, 1985.

M. Phelan. Ford speeds changeovers in engine production. *Knight Ridder Tribune Business News, Washington*, November 6, 2002.

H. Pirkul and V. Jayaraman. Production, transportation, and distribution planning in a multi-commodity tri-echelon system. *Transportation Science*, 30(4):291–302, 1996.

Y. Pochet and L. Wolsey. Algorithms and reformulations for lot sizing problems. CORE discussion paper 9427, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE), 1995.

Y. Pochet and L. Wolsey. *Production Planning by Mixed Integer Programming*. Springer-Verlag, New York, 2006.

E. L. Porteus. The newsvendor problem. In D. Chhajed and T. J. Lowe, editors, *Building Intuition: Insights From Basic Operations Management Models and Principles*, chapter 7, pages 115–134. Springer, New York, 2008.

J.-Y. Potvin. State-of-the art review—evolutionary algorithms for vehicle routing. *INFORMS Journal on Computing*, 21(4):518–548, 2009.

W. B. Powell. A local improvement heuristic for the design of less-than-truckload motor carrier networks. *Transportation Science*, 20(4):246–257, 1986.

G. P. Prastacos. Blood inventory management: An overview of theory and practice. *Management Science*, 30(7):777–800, 1984.

R. C. Prim. Shortest connection networks and some generalizations. *Bell System Technical Journal*, 36(6):1389–1401, 1957.

C. Prodhon and C. Prins. A survey of recent research on location-routing problems. *European Journal of Operational Research*, 238(1):1–17, 2014.

L. Qi, Z.-J. M. Shen, and L. V. Snyder. The effect of supply disruptions on supply chain design decisions. *Transportation Science*, 44(2):274–289, 2010.

W. Qi, Y. Liang, and Z.-J. M. Shen. Joint planning of energy storage and transmission for wind energy generation. *Operations Research*, 63(6):1280–1293, 2015.

T. K. Ralphs, L. Kopman, W. R. Pulleyblank, and L. E. Trotter Jr. On the capacitated vehicle routing problem. *Mathematical Programming*, 94:343–359, 2003.

M. Ramming. *Network Knowledge and Route Choice*. Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, 2001.

J. Reese. Solution methods for the p-median problem: An annotated bibliography. *Networks*, 48(3): 125–142, 2006.

C. Rego. A subpath ejection method for the vehicle routing problem. *Management Science*, 44(10): 1447–1459, 1998.

C. Rego and C. Roucairol. A parallel tabu search algorithm using ejection chains for the vehicle routing problem. In I. H. Osman and J. P. Kelly, editors, *Meta-Heuristics: Theory and Applications*, pages 661–675. Springer US, New York, 1996.

G. Reinelt. TSPLIB—a traveling salesman problem library. *ORSA Journal on Computing*, 3(4): 376–384, 1991.

J. Renaud, F. F. Boctor, and G. Laporte. An improved petal heuristic for the vehicle routeing problem. *Journal of the Operational Research Society*, 47(2):329–336, 1996.

C. ReVelle. The maximum capture or "sphere of influence" location problem: Hotelling revisited on a network. *Journal of Regional Science*, 26(2):343–358, 1986.

C. Revelle, D. Marks, and J. C. Liebman. An analysis of private and public sector location models. *Management Science*, 16(11):692–707, 1970.

C. ReVelle, M. Scholssberg, and J. Williams. Solving the maximal covering location problem with heuristic concentration. *Computers & Operations Research*, 35(2):427–435, 2008a.

C. S. ReVelle and R. W. Swain. Central facilities location. *Geographical Analysis*, 2:30–42, 1970.

C. S. ReVelle, H. A. Eiselt, and M. S. Daskin. A bibliography for some fundamental problem categories in discrete location science. *European Journal of Operational Research*, 184(3): 817–848, 2008b.

L. W. Robinson. Optimal and approximate policies in multiperiod, multilocation inventory models with transshipments. *Operations Research*, 38(2):278–295, 1990.

Y. Rochat and E. D. Taillard. Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics*, 1(1):147–167, 1995.

R. Romero, R. A. Gallego, and A. Monticelli. Transmission system expansion planning by simulated annealing. In *Proceedings of Power Industry Computer Applications Conference*, pages 278–283, Salt Lake City, UT, 1995. IEEE.

Y. Rong, Z. Atan, and L. V. Snyder. Heuristics for base-stock levels in multi-echelon distribution networks. *Production and Operations Management*, 26(9):1760–1777, 2017a.

Y. Rong, L. V. Snyder, and Z.-J. M. Shen. Bullwhip and reverse bullwhip effects under the rationing game. *Naval Research Logistics*, 64(3):203–216, 2017b.

D. J. Rosenkrantz, R. E. Stearns, and P. M. Lewis II. An analysis of several heuristics for the traveling salesman problem. *SIAM Journal on Computing*, 6(3):563–581, 1977.

K. E. Rosing, C. S. Revelle, and H. Rosing-Vogelaar. The p-median and its linear programming relaxation: An approach to large problems. *Journal of the Operational Research Society*, 30 (9):815–823, 1979.

K. Rosling. Optimal inventory policies for assembly systems under random demands. *Operations Research*, 37(4):565–579, 1989.

S. M. Ross. *Stochastic Processes*. John Wiley & Sons, New York, 2nd edition, 1996.

R. Roundy. 98%-effective integer-ratio lot-sizing for one-warehouse multi-retailer systems. *Management Science*, 31(11):1416–1430, 1985.

P. Rusmevichientong, Z.-J. M. Shen, and D. B. Shmoys. Dynamic assortment optimization with a multinomial logit choice model and capacity constraint. *Operations Research*, 58(6):1666–1680, 2010.

R. Russell and W. Igo. An assignment routing problem. *Networks*, 9(1):1–17, 1979.

R. A. Russell. An effective heuristic for the m-tour traveling salesman problem with some side conditions. *Operations Research*, 25(3):517–524, 1977.

J. K. Ryan. *Analysis of Inventory Models with Limited Demand Information*. Ph.D. dissertation, Northwestern University, Department of Industrial Engineering and Management Sciences, Evanston, IL, 1997.

S. Sahni and T. Gonzalez. P-complete approximation problems. *Journal of the ACM*, 23(3):555–565, 1976.

M. W. P. Savelsbergh. Local search in routing problems with time windows. *Annals of Operations Research*, 4(1):285–305, 1985.

S. Savin and C. Terwiesch. Optimal product launch times in a duopoly: Balancing life-cycle revenues with product cost. *Operations Research*, 53(1):26–47, 2005.

H. Scarf. The optimality of (s, S) policies in the dynamic inventory problem. In K. Arrow, S. Karlin, and P. Suppes, editors, *Mathematical Methods in the Social Sciences*, chapter 13. Stanford University Press, Stanford, CA, 1960.

F. Schalekamp, D. P. Williamson, and A. van Zuylen. 2-matchings, the traveling salesman problem, and the subtour LP: A proof of the Boyd-Carr conjecture. *Mathematics of Operations Research*, 39(2):403–417, 2014.

D. A. Schilling. Dynamic location modeling for public-sector facilities: A multicriteria approach. *Decision Sciences*, 11(4):714–724, 1980.

G. M. Schmidt and C. T. Druehl. Changes in product attributes and costs as drivers of new product diffusion and substitution. *Production and Operations Management*, 14(3):272–285, 2005.

A. J. Schmitt, L. V. Snyder, and Z.-J. M. Shen. Inventory systems with stochastic demand and supply: Properties and approximations. *European Journal of Operational Research*, 206(2):313–328, 2010.

A. J. Schmitt, S. A. Sun, L. V. Snyder, and Z.-J. M. Shen. Centralization versus decentralization: Risk pooling, risk diversification, and supply uncertainty in a one-warehouse multiple-retailer system. *OMEGA*, 52:201–212, 2015.

L. Schrage. Implicit representation of variable upper bounds in linear programming. *Mathematical Programming Studies*, 4:118–132, 1978.

A. J. Scott. Dynamic location-allocation systems: Some basic planning strategies. *Environment and Planning A*, 3(1):73–82, 1971.

R. Serfozo and S. Stidham. Semi-stationary clearing processes. *Stochastic Processes and Their Applications*, 6(2):165–178, 1978.

D. Serra and V. Marianov. The p-median problem in a changing network: The case of Barcelona. *Location Science*, 6:383–394, 1998.

D. Serra and C. ReVelle. Market capture by two competitors: The preemptive location problem. *Journal of Regional Science*, 34(4):549–561, 1994.

D. Serra, S. Ratick, and C. ReVelle. The maximum capture problem with uncertainty. *Environment and Planning B*, 23:49–59, 1996.

M. I. Shamos. Geometric complexity. In *Proceedings of Seventh Annual ACM Symposium on Theory of Computing*, pages 224–233, New York, 1975. ACM.

M. I. Shamos and D. Hoey. Closest-point problems. In *Proceedings of the 16th Annual Symposium on Foundations of Computer Science*, pages 151–162, 1975.

K. H. Shang and J.-S. Song. Newsvendor bounds and heuristic for optimal policies in serial supply chains. *Management Science*, 49(5):618–638, 2003.

Z.-J. M. Shen. Integrated supply chain design models: A survey and future research directions. *Journal of Industrial and Management Optimization*, 3(1):1–27, 2007.

Z.-J. M. Shen, C. R. Coullard, and M. S. Daskin. A joint location-inventory model. *Transportation Science*, 37(1):40–55, 2003.

Z.-J. M. Shen, R. L. Zhan, and J. Zhang. The reliable facility location problem: Formulations, heuristics, and approximation algorithms. *INFORMS Journal on Computing*, 23(3):470–482, 2011.

C. C. Sherbrooke. METRIC: A multi-echelon technique for recoverable item control. *Operations Research*, 16(1):122–141, 1968.

C. C. Sherbrooke. Waiting time in an $(S-1, S)$ inventory system: Constant service time case. *Operations Research*, 23(4):819–820, 1975.

D. R. Shier. A min-max theorem for p-center problems on a tree. *Transportation Science*, 11(3): 243–252, 1977.

D. R. Shier and P. M. Dearing. Optimal locations for a class of nonlinear, single-facility location problems on a network. *Operations Research*, 31(2):292–303, 1983.

D. B. Shmoys and D. P. Williamson. Analyzing the Held-Karp TSP bound: A monotonicity property with application. *Information Processing Letters*, 35:281–285, 1990.

J. Shu, C.-P. Teo, and Z.-J. M. Shen. Stochastic transportation-inventory network design problem. *Operations Research*, 53(1):48–60, 2005.

E. A. Silver and D. P. Bischak. The exact fill rate in a periodic review base stock system under normally distributed demand. *OMEGA*, 39(3):346–349, 2011.

E. A. Silver, D. F. Pyke, and R. Peterson. *Inventory Management and Production Planning and Scheduling*. John Wiley & Sons, New York, 3rd edition, 1998.

D. Simchi-Levi and Y. Wei. Understanding the performance of the long chain and sparse designs in process flexibility. *Operations Research*, 60(5):1125–1141, 2012.

D. Simchi-Levi, X. Chen, and J. Bramel. *The Logic of Logistics*. Springer, New York, 3rd edition, 2013.

D. Simchi-Levi, W. Schmidt, Y. Wei, P. Y. Zhang, K. Combs, Y. Ge, O. Gusikhin, M. Sanders, and D. Zhang. Identifying risks and mitigating disruptions in the automotive supply chain. *Interfaces*, 45(5):375–390, 2015.

K. F. Simpson, Jr. In-process inventories. *Operations Research*, 6(6):863–873, 1958.

L. V. Snyder. Facility location under uncertainty: A review. *IIE Transactions*, 38(7):537–554, 2006.

L. V. Snyder. Covering problems. In H. A. Eiselt and V. Marianov, editors, *Foundations of Location Analysis*, chapter 6, pages 109–135. Springer-Verlag, Berlin, 2011.

L. V. Snyder. A tight approximation for a continuous-review inventory model with supplier disruptions. *International Journal of Production Economics*, 155:91–108, 2014.

L. V. Snyder. Inventory and supply chain optimization. In T. Terlaky, M. Anjos, and S. Ahmed, editors, *Advances and Trends in Optimization with Engineering Applications*, chapter 33, pages 439–455. SIAM, Philadelphia, 2017.

L. V. Snyder and M. S. Daskin. Reliability models for facility location: The expected failure cost case. *Transportation Science*, 39(3):400–416, 2005.

L. V. Snyder and M. S. Daskin. Models for reliable supply chain network design. In A. T. Murray and T. H. Grubesic, editors, *Critical Infrastructure: Reliability and Vulnerability*, chapter 13, pages 257–289. Springer-Verlag, New York, 2007.

L. V. Snyder and Z.-J. M. Shen. Supply and demand uncertainty in multi-echelon supply chains. Technical report, Lehigh University, Bethlehem, PA, 2006.

L. V. Snyder and B. T. Tomlin. On the value of a threat advisory system for managing supply chain disruptions. Technical report, Lehigh University, Bethlehem, PA, 2007.

L. V. Snyder, M. P. Scaparra, M. S. Daskin, and R. L. Church. Planning for disruptions in supply chain networks. In M. P. Johnson, B. Norman, and N. Secomandi, editors, *Tutorials in Operations Research*, chapter 9, pages 234–257. INFORMS, Hanover, MD, 2006.

L. V. Snyder, Z. Atan, P. Peng, Y. Rong, A. Schmitt, and B. Sinsoysal. OR/MS models for supply chain disruptions: A review. *IIE Transactions*, 48(2):89–109, 2016.

M. M. Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265, 1987.

J.-S. Song and P. H. Zipkin. Inventory control with information about supply conditions. *Management Science*, 42(10):1409–1419, 1996.

J. J. Spengler. Vertical integration and antitrust policy. *Journal of Political Economy*, 58(4):347–352, 1950.

J. D. Sterman. Modeling managerial behavior: Misperceptions of feedback in a dynamic decision making experiment. *Management Science*, 35(3):321–339, 1989.

W. R. Stewart. A computationally efficient heuristic for the traveling salesman problem. In *Proceedings of the 13th Annual Meetings of S.E. TIMS*, pages 75–85, 1977.

F. Sultan, J. U. Farley, and D. R. Lehmann. A meta-analysis of applications of diffusion models. *Journal of Marketing Research*, 27(1):70–77, 1990.

T. Sundstrom. *Mathematical Reasoning: Writing and Proof.* Prentice Hall, Upper Saddle River, NJ, 2nd edition, 2006.

R. W. Swain. A parametric decomposition approach for the solution of uncapacitated location problems. *Management Science*, 21(2):189–198, 1974.

J. M. Swaminathan and S. R. Tayur. Managing broader product lines through delayed differentiation using vanilla boxes. *Management Science*, 44(12):S161–S172, 1998.

E. W. Taft. The most economical production lot. *Iron Age*, 101:1410–1412, 1918.

G. Tagaras. Effects of pooling on the optimization and service levels of two-location inventory systems. *IIE Transactions*, 21:250–257, 1989.

G. Tagaras. Pooling in multi-location periodic inventory distribution systems. *OMEGA*, 27:39–59, 1999.

E. Taillard. Parallel iterative search methods for vehicle routing problems. *Networks*, 23(8):661–673, 1993.

A. C. Tamhane and D. D. Dunlop. *Statistics and Data Analysis: From Elementary to Intermediate*. Prentice Hall, Upper Saddle River, NJ, 1999.

C. Tan and J. E. Beasley. A heuristic algorithm for the period vehicle routing problem. *OMEGA*, 12 (5):497–504, 1984.

B. C. Tansel. Discrete center problems. In H. A. Eiselt and V. Marianov, editors, *Foundations of Location Analysis*, chapter 5, pages 79–106. Springer-Verlag, Berlin, 2011.

B. C. Tansel, R. L. Francis, T. J. Lowe, and M. Chen. Duality and distance constraints for the nonlinear p-center problem and covering problem on a tree network. *Operations Research*, 30 (4):725–744, 1982.

M. B. Teitz and P. Bart. Heuristic methods for estimating the generalized vertex median of a weighted graph. *Operations Research*, 16(5):955–961, 1968.

C.-P. Teo and J. Shu. Warehouse-retailer network design problem. *Operations Research*, 52(3): 396–408, 2004.

C.-P. Teo, J. Ou, and M. Goh. Impact on inventory costs with consolidation of distribution centers. *IIE Transactions*, 33(2):99–110, 2001.

R. H. Teunter. Note on the fill rate of single-stage general periodic review inventory systems. *Operations Research Letters*, 37(1):67–68, 2009.

P. M. Thompson and H. N. Psaraftis. Cyclic transfer algorithm for multivehicle routing and scheduling problems. *Operations Research*, 41(5):935–946, 1993.

B. T. Tomlin. On the value of mitigation and contingency strategies for managing supply-chaindisruption risks (unabridged version). Working paper, Kenan-Flagler Business School, UNC-Chapel Hill, April 2005.

B. T. Tomlin. On the value of mitigation and contingency strategies for managing supply chain disruption risks. *Management Science*, 52(5):639–657, 2006.

H. Topaloglu. A tighter variant of Jensen's lower bound for stochastic programs and separable approximations to recourse functions. *European Journal of Operational Research*, 199(2): 315–322, 2009.

C. Toregas and C. ReVelle. Optimal location under time or distance constraints. *Papers of the Regional Science Association*, 28:133–143, 1972.

C. Toregas, R. Swain, C. ReVelle, and L. Bergman. The location of emergency service facilities. *Operations Research*, 19(6):1363–1373, 1971.

P. Toth and D. Vigo. An exact algorithm for the vehicle routing problem with backhauls. *Transportation Science*, 31:372–385, 1997.

P. Toth and D. Vigo. A heuristic algorithm for the symmetric and asymmetric vehicle routing problems with backhauls. *European Journal of Operational Research*, 113:528–543, 1999.

P. Toth and D. Vigo, editors. *The Vehicle Routing Problem.* Society for Industrial and Applied Mathematics, Philadelphia, 2001a.

P. Toth and D. Vigo. Branch-and-bound algorithms for the capacitated VRP. In P. Toth and D. Vigo, editors, *The Vehicle Routing Problem*, chapter 2, pages 29–51. Society for Industrial and Applied Mathematics, Philadelphia, 2001b.

P. Toth and D. Vigo. VRP with backhauls. In P. Toth and D. Vigo, editors, *The Vehicle Routing Problem*, chapter 8, pages 195–224. Society for Industrial and Applied Mathematics, Philadelphia, 2001c.

P. Toth and D. Vigo. The granular tabu search and its application to the vehicle-routing problem. *INFORMS Journal on Computing*, 15(4):333–346, 2003.

Toyota Motor Corporation. Annual report 2012, 2012. URL `https://www.toyota-global.com/pages/contents/investors/ir_library/annual/pdf/2012/ar12_e.pdf`.

K. Train. A validation test of a disaggregate mode choice model. *Transportation Research*, 12: 167–174, 1978.

K. Train, D. McFadden, and M. Ben-Akiva. The demand for local telephone service: A fully discrete model of residential calling patterns and service choices. *RAND Journal of Economics*, 18(1): 109–123, 1987.

K. E. Train. *Discrete Choice Methods with Simulation*. Cambridge University Press, New York, NY, 2nd edition, 2009.

A. A. Tsay. The quantity flexibility contract and supplier-customer incentives. *Management Science*, 45:1339–1358, 1999.

G. L. Vairaktarakis and P. Kouvelis. Incorporation dynamic aspects and uncertainty in 1-median location problems. *Naval Research Logistics*, 46(2):147–168, 1999.

A. J. Vakharia and A. Yenipazarli. Managing supply chain disruptions. *Foundations and Trends in Technology, Information and Operations Management*, 2(4):243–325, 2008.

J. Van Biesebroeck. Complementarities in automobile production. *Journal of Applied Econometrics*, 22(7):1315–1345, 2007.

A. Van Breedam. An analysis of the effect of local improvement operators in genetic algorithms and simulated annealing for the vehicle routing problem. RUCA working paper 96/14, University of Antwerp, Belgium, 1996.

G. J. van Houtum, K. Inderfurth, and W. Zijm. Materials coordination in stochastic multi-echelon systems. *European Journal of Operational Research*, 95:1–23, 1996.

T. J. Van Roy. A cross decomposition algorithm for capacitated facility location. *Operations Research*, 34(1):145–163, 1986.

T. J. Van Roy and D. Erlenkotter. A dual-based procedure for dynamic facility location. *Management Science*, 28(10):1091–1105, 1982.

A. F. Veinott, Jr. Optimal policy for a multi-product, dynamic, nonstationary inventory problem. *Management Science*, 12(3):206–222, 1965.

A. F. Veinott, Jr. On the optimality of (s, S) inventory policies: New conditions and a new proof. *SIAM Journal on Applied Mathematics*, 14(5):1067–1083, 1966.

A. F. Veinott, Jr. and H. M. Wagner. Computing optimal (s, S) inventory policies. *Management Science*, 11(5):525–552, 1965.

D. J. Velleman. *How to Prove It: A Structured Approach*. Cambridge University Press, New York, NY, 2nd edition, 2006.

K. Venkatesh, V. Ravi, A. Prinzie, and D. Van den Poel. Cash demand forecasting in ATMs by clustering and neural networks. *European Journal of Operational Research*, 232(2):383–392, 2014.

M. Ventosa, Á. Baíllo, A. Ramos, and M. Rivier. Electricity market modeling trends. *Energy Policy*, 33(7):897–913, 2005.

T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins. Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. *European Journal of Operational Research*, 231(1):1–21, 2013.

P. VonAchen, K. Smilowitz, M. Raghavan, and R. Feehan. Optimizing community healthcare coverage in remote Liberia. *Journal of Humanitarian Logistics and Supply Chain Management*, 6(3): 352–371, 2016.

A. Wagelmans, S. V. Hoesel, and A. Kolen. Economic lot sizing: An $O(n \log n)$ algorithm that runs in linear time in the Wagner-Whitin case. *Operations Research*, 40(S1):S145–S156, 1992.

H. M. Wagner and T. M. Whitin. Dynamic version of the economic lot size model. *Management Science*, 5(1):89–96, 1958.

A. Warburton. Worst-case analysis of some convex hull heuristics for the Euclidean travelling salesman problem. *Operations Research Letters*, 13(1):37–42, 1993.

P. Wark and J. Holt. A repeated matching heuristic for the vehicle routeing problem. *Journal of the Operational Research Society*, 45(10):1156–1167, 1994.

J. R. Weaver and R. L. Church. Computational procedures for location problems on stochastic networks. *Transportation Science*, 17(2):168–180, 1983.

G. O. Wesolowsky. Dynamic facility location. *Management Science*, 19(11):1241–1248, 1973.

R. A. Whitaker. A fast algorithm for the greedy interchange of large-scale clustering and median location problems. *INFOR*, 21(2):95–108, 1983.

T. M. Whitin. *The Theory of Inventory Management*. Princeton University Press, Princeton, NJ, 1953.

W. E. Wilhelm. A technical review of column generation in integer programming. *Optimization and Engineering*, 2(2):159–200, 2001.

P. R. Winters. Forecasting sales by exponentially weighted moving averages. *Management Science*, 6(3):324–342, 1960.

L. A. Wolsey. Heuristic analysis, linear programming and branch and bound. *Mathematical Programming Study*, 13:121–134, 1980.

R. T. Wong. A dual ascent approach for Steiner tree problems on a directed graph. *Mathematical Programming*, 28(3):271–287, 1984.

World Health Organization. Influenza (seasonal): Fact sheet, January 2018. URL `http://www.who.int/mediacentre/factsheets/fs211/en`.

A. Wren. *Computers in Tranport Planning and Operation*. Ian Allan, London, 1971.

A. Wren and A. Holliday. Computer scheduling of vehicles from one or more depots to a number of delivery points. *Journal of the Operational Research Society*, 23:333–344, 1972.

P. D. Wright, M. J. Liberatore, and R. L. Nydick. A survey of operations research models and applications in homeland security. *Interfaces*, 36(6):514–529, 2006.

O. Q. Wu and Y. Ouyang. Supply chain design and optimization with applications in the energy industry. In T. Terlaky, M. Anjos, and S. Ahmed, editors, *Advances and Trends in Optimization with Engineering Applications*, chapter 34, pages 457–468. SIAM, Philadelphia, 2017.

S. D. Wu, B. Aytac, R. Berger, and C. Armbruster. Managing short life-cycle technology products for Agere systems. *Interfaces*, 36(3):234–247, 2006.

S. D. Wu, K. G. Kempf, M. O. Atan, B. Aytac, S. A. Shirodkar, and A. Mishra. Improving new-product forecasting at Intel Corporation. *Interfaces*, 40(5):385–396, 2010.

J. Xu and J. P. Kelly. A network flow-based tabu search heuristic for the vehicle routing problem. *Transportation Science*, 30(4):379–393, 1996.

P. Yadav, D. M. Miller, C. P. Schmidt, and R. Drake. McGriff Treading Company implements service contracts with shared savings. *Interfaces*, 33(6):18–29, 2003.

C. A. Yano and H. L. Lee. Lot sizing with random yield: A review. *Operations Research*, 43(2): 311–334, 1995.

P. Yellow. A computational modification to the savings method of vehicle scheduling. *Operational Research Quarterly*, 21:281–283, 1970.

H. Younies and H. A. Eiselt. Sequential location models. In H. A. Eiselt and V. Marianov, editors, *Foundations of Location Analysis*, chapter 8, pages 163–178. Springer-Verlag, Berlin, 2011.

W. I. Zangwill. A deterministic multi-period production scheduling model with backlogging. *Management Science*, 13(1):105–119, 1966.

J. Zhang and J. Zhang. Fill rate of single-stage general periodic review inventory systems. *Operations Research Letters*, 35(4):503–509, 2007.

X. Zhang. The impact of forecasting methods on the bullwhip effect. *International Journal of Production Economics*, 88:15–27, 2004.

H. Zhao. Supply chain optimization in healthcare. In T. Terlaky, M. Anjos, and S. Ahmed, editors, *Advances and Trends in Optimization with Engineering Applications*, chapter 35, pages 469–478. SIAM, Philadelphia, 2017.

Y. Zheng. A simple proof for the optimality of (s, S) policies for infinite-horizon inventory problems. *Journal of Applied Probability*, 28:802–810, 1991.

Y.-S. Zheng. On properties of stochastic inventory systems. *Management Science*, 38(1):87–103, 1992.

Y.-S. Zheng and A. Federgruen. Finding optimal (s, S) policies is about as simple as evaluating a single policy. *Operations Research*, 39(4):654–665, 1991.

Y.-S. Zheng and A. Federgruen. Errata: Finding optimal (s, S) policies is about as simple as evaluating a single policy. *Operations Research*, 40(1):192, 1992.

D. Zhou, L. C. Leung, and W. P. Pierskalla. Inventory management of platelets in hospitals: Optimal inventory policy for perishable products with regular and optional expedited replenishments. *Manufacturing & Service Operations Management*, 13(4):420–438, 2011.

Y. Zhou, A. Scheller-Wolf, N. Secomandi, and S. Smith. Managing wind-based electricity generation in the presence of storage and transmission capacity. Working paper, Tepper School of Business, Carnegie Mellon University, Pittsburgh, 2018.

P. Ziobro. As web sales spike, retailers scramble to ship from stores. *Wall Street Journal*, December 1, 2016.

P. H. Zipkin. Inventory service-level measures: Convexity and approximation. *Management Science*, 32(8):975–981, 1986a.

P. H. Zipkin. Stochastic leadtimes in continuous-time inventory models. *Naval Research Logistics Quarterly*, 33(4):763–774, 1986b.

P. H. Zipkin. *Foundations of Inventory Management*. Irwin/McGraw-Hill, New York, 2000.

P. H. Zipkin. Old and new methods for lost-sales inventory systems. *Operations Research*, 56(5): 1256–1263, 2008a.

P. H. Zipkin. On the structure of lost-sales inventory models. *Operations Research*, 56(4):937–944, 2008b.

SUBJECT INDEX

 Fundamentals of Supply Chain Theory, Second Edition. Lawrence V. Snyder and Zuo-Jun Max Shen.

Companion website: www.wiley.com/go/Snyder/SupplyChainTheory

AUTHOR INDEX